



TESIS TE 142599

PENINGKATAN KAPASITAS PANGGILAN PADA SERVER VOIP BERBASIS EMBEDDED SYSTEM DENGAN METODE NATIVE BRIDGING

Edi Prihartono
2214206005

DOSEN PEMBIMBING
Dr. Ir. Achmad Affandi, DEA.
Dr. Istas Pratomo, ST., MT.

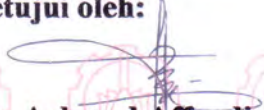


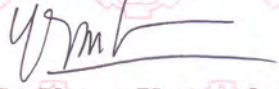


PROGRAM MAGISTER
BIDANG KEAHLIAN TELEMATIKA
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2016

**Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Teknik (MT)
di
Institut Teknologi Sepuluh Nopember**

**Oleh:
Edi Prihartono
NRP. 2214206005**

**Tanggal ujian : 22 Juni 2016
Periode wisuda : September 2016**

Disetujui oleh:

1.  **Dr. Ir. Achmad Affandi, DEA.** (Pembimbing I)
NIP. 196510141990021001
2.  **Dr. Istas Pratomo ST., MT.** (Pembimbing II)
NIP. 197903252003121001
3.  **Dr. Surya Sumpeno, ST., M.Sc.** (Penguji)
NIP. 196906131997021003
4.  **Dr. Ir. Yoyon Kusnendar Suprpto, M.Sc.** (Penguji)
NIP. 195409251978031001
5.  **Mochamad. Hariadi, ST., M.Sc., Ph.D.** (Penguji)
NIP. 196912091997031002
6.  **Dr. I Ketut Eddy Purnama, ST., MT.** (Penguji)
NIP. 196907301995121001

Direktur Program Pascasarjana,



Prof. Ir. Djauhar Manfaat, M.Sc., Ph.D.
NIP. 196012021987011001

PERNYATAAN KEASLIAN TESIS

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan tesis saya dengan judul “**PENINGKATAN KAPASITAS PANGGILAN PADA SERVER VOIP BERBASIS EMBEDDED SYSTEM DENGAN METODE NATIVE BRIDGING**” adalah benar hasil karya intelektual mandiri, disesuaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan.

Surabaya, 12 Juli 2016

(Edi Prihartono)
NRP.2214206005

!-- Halaman ini sengaja di kosongkan --!

Peningkatan Kapasitas Panggilan pada Server VoIP Berbasis Embedded System dengan Metode Native Bridging

NamaMahasiswa : Edi Prihartono
NRP : 22142030005
Pembimbing : Dr. Ir. Achmad Affandi, DEA.
: Dr. Istas Pratomo, ST., MT.

ABSTRAK

Komunikasi VoIP telah berkembang dengan pesat. Pada penelitian sebelumnya. telah dilakukan desain komunikasi bergerak nirkabel berbasis OpenWRT dengan perancangan prototype awal yaitu WiFi BTS menggunakan AP-Router yang telah dimodifikasi sebagai Server VoIP. Dimana Server VoIP pada penelitian sebelumnya dalam tahap perancangan masih perlu pengembangan dan masih terdapat beberapa keterbatasan, salah satunya adalah masalah kapasitas jumlah panggilan yang sedikit , berjumlah 22 client (11 pasang).

Sehingga perlu adanya cara untuk menambah jumlah kapasitas panggilan yang dapat di layani oleh Server VoIP. Pada penelitian ini kami usulkan suatu metode penanganan tranfer media data yaitu metode Native Bridging. Dengan menggunakan Native Bridging kapasitas jumlah panggilan yang mampu dilayani oleh Server VoIP berjumlah 26 client(13 pasang), terjadi peningkatan kapasitas panggilan sebesar 18,18%. Sedangkan kemampuan layanan panggilan dengan multi node untuk menambah jarak jangkauan, hanya mampu melayani panggilan menjadi 20 client atau 10 pasang, terjadi penurunan kapasitas jumlah panggilan sebesar 23,08%.

Kata Kunci ; VoIP, OpenWRT, Peningkatan Kapasitas, Native Bridging

!-- Halaman ini sengaja di kosongkan --!

The Increased in Call Capacity on VoIP Server Based Embedded System with Bridging Native Method

Name : Edi Prihartono
Student Identity Number : 22142030005
Supervisor : Dr. Ir. Achmad Affandi, DEA
: Dr. Istas Pratomo, ST, MT

ABSTRAC

Communications VoIP has grown rapidly. In previous research, has done design-based mobile wireless communications OpenWRT with the initial prototype design is WiFi base stations using the AP-Router that has been modified as a VoIP server. Where VoIP Server on previous research in the design stage and still need development there are still some limitations, one of which is a capacity problem with the number of calls that a little bit number 22 client (11 pairs).

So it needs a way to increase the number of call capacity that can be examined by voip server. In this study, we propose a method of handling the transfer of media data that Bridging Native methods. Bridging Native capacity using the number of calls able to be serviced by a server voip benjumlah 26 client (13 pairs), an increase in call capacity amounted to 18.18%. While the ability of a service call with a multi node to add distance range, only serve client calls to 20 or 10 pairs, a decline in the number of call capacity by 23.08%

Keywords; VoIP, OpenWRT, Increased Capacity, Native Bridging

!!-- Halaman ini sengaja di kosongkan --!!

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah Yang Maha Kuasa, atas rahmat dan hidayahnya, sehingga penulis dapat menyelesaikan penelitian “**PENINGKATAN KAPASITAS PANGGILAN PADA SERVER VOIP BERBASIS EMBEDDED SYSTEM DENGAN METODE NATIVE BRIDGING**” yang tertulis pada buku tesis ini. Tesis ini disusun sebagai salah satu syarat akademik kelulusan Program Studi Magister, Bidang Minat TELEMATIKA, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS), Surabaya.

Dalam proses penyusunan buku tesis ini, penulis mendapatkan bantuan dari berbagai pihak, sehingga dapat terselesaikan. Atas bantuannya kami sampaikan terimah kasih kepada:

1. Bapak ibu dan bapak ibu mertua yang telah memberikan doa untuk kelancaran selama studi.
2. Kepada istriku tercinta Kartika Udiyaningsih, anak-anakku tersayang Naufal Ikbar Nur Ramadhan dan Izzudin Syahrur Nur Ramadhan, yang selalu memberikan doa dan dorongan semangat untuk menyelesaikan studi.
3. Dr. Ir. Achmad Affandi, DEA dan Dr. Istas Pratomo, ST., MT., selaku pembimbing, yang selalu memberikan masukan dan dorongan semangat untuk menyelesaikan penelitian ini
4. Dr. Surya Sumpeno, ST., M.Sc., Dr. Ir. Yoyon Kusnendar Suprpto, M.Sc., Moch. Hariyadi ,ST., M.Sc., Ph.D dan Dr. I Ketut Eddy Purnama, ST., MT., sebagai penguji sidang tesis yang dengan teliti dan memberikan masukan sehingga penelitian ini lebih baik.
5. Rekan-rekan kerja Universitas DR.Soetomo Surabaya, yang selalu memberikan dorongan semangat untuk menyelesaikan studi.
6. Rekan-rekan Telematika Angkatan 2014, yang selalu kompak selama studi sukses untuk kalian.

7. Rekan-rekan Lab.B301 Jaringan Telekomunikasi terima kasih atas kerjasamanya.

Dan pihak-pihak lain yang ikut membantu, tetapi tidak bisa saya sebutkan satu persatu. Semoga Allah membalas amal baik ibadah mereka....Amin.

Penulis menyadari dalam penyusunan buku tesis ini, masih jauh dari sempurna, penulis menerima masukan kritikan dan saran yang membangun, yang menjadikan buku tesis ini menjadi lebih baik

Surabaya,12 Juli 2016

Penulis

DAFTAR ISI

	Halaman
LEMBAR PENGESAHAN.....	i
ABSTRAK	iii
ABSTRAC	v
KATA PENGANTAR	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xv
DAFTAR LAMPIRAN	xvii
BAB 1 <u>P</u> ENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	2
1.5 Manfaat Penelitian.....	2
1.6 Sistematika Penulisan.....	3
BAB 2 <u>K</u> AJIAN PUSTAKA.....	5
2.1 Manet Adhoc Network(MANET)	5
2.2 Routing Protokol Pada Manet	5
2.2.1 Routing Protokol Reaktif.....	6
2.2.2 Routing Protokol Proaktif.....	6
2.2.3 Routing Protokol Hybrid	6
2.2.4 Routing Protokol OLSR.....	6
2.2.5 OLSR On Daemon.....	7

2.3	Voice Internet Protokol (VoIP)	8
2.3.1	Protokol Persinyalan	9
2.3.2	Protokol Media Tranfer	10
2.3.3	Coder dan Decoder (Codec) VoIP	12
2.3.4	<i>Quality Of Service</i> (QOS)	14
2.4	Operating Sistem untuk Router Wireless	16
2.4.1	Sistem pada Server VoIP	16
2.4.2	TP Link MR-3020	17
2.4.3	Sistem pada VoiP Client	17
2.5	Bridging	18
2.6	Aplikasi Pengukuran.....	19
2.7	Penelitian Terkait.....	20
BAB 3	METODOLOGI PENELITIAN	21
3.1	Rancangan Penelitian.....	21
3.2	Gambaran Umum Sistem.....	21
3.2.1	Tahapan Instalasi Sistem OpenWRT pada Router	22
3.2.2	Tahapan Instalasi Open VoIP Sistem.....	23
3.3	Skenario Pengujian	23
3.3.1	Pengujian Panggilan pada Sistem	24
3.3.2	Pengujian Kapasitas Maksimum Panggilan Per Node	26
3.3.3	Pengujian Kualitas Koneksi dan Kapasitas Panggilan Antar Node .	26
BAB 4	HASIL PENELITIAN DAN PEMBAHASAN.....	29
4.1	Implementasi Rancangan.....	29
4.1.1	Instalasi Sistem OpenWRT pada Router.....	29
4.1.2	Konfigurasi Open Voice Sistem.....	31
4.1.3	Install dan Konfigurasi Extroot.....	31

4.1.4 Instalasi Server VoIP	33
4.1.5 Konfigurasi Native Bridging	34
4.2 Data Hasil Pengujian	35
4.2.1 Data Pengujian Panggilan pada Sistem	36
4.2.2 Data Pengujian Kapasitas Maksimum Panggilan Per Node	37
4.2.3 Data Pengujian Koneksi dan Kapasitas antar Node.....	39
4.3 Analisa Data Pengujian	40
4.3.1 Analisa Data Pengujian Call pada Sistem.....	40
4.3.2 Analisa Data Kapasitas Maksimum Panggilan Per Node	42
4.3.3 Analisa Data Pengujian Koneksi dan Kapasitas Panggilan Antar Node.....	46
BAB 5_KESIMPULAN DAN SARAN	51
5.1 Kesimpulan Penelitian.....	51
5.2 Saran.....	52
DAFTAR PUSTAKA	53
HALAMAN LAMPIRAN	55
7.1 Konfigurasi Penambahan Kapasitas memori.....	55
7.2. Konfigurasi Native Bridging	55
7.3. Konfigurasi OLSR.....	64
RIWAYAT HIDUP.....	67

!-- Halaman ini sengaja di kosongkan --!

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Klasifikasi protokol pada MANet	5
Gambar 2.2 Perbandingan Sistem broadcasting (a) Link state broadcast (b) MPR broadcast	7
Gambar 2.3 Cara Kerja VoIP	8
Gambar 2.4 Hirarki Protokol VoIP	9
Gambar 2.5 Saluran Bridging	18
Gambar 2.6 Skema Native Bridging	19
Gambar 3.1 Digram Flowchat Sistem	21
Gambar 3.2 Arsitektur Jaringan VoiP	22
Gambar 3.3 Call antar client	25
Gambar 3.4 Jarak panggilan antara node dengan client.....	25
Gambar 3.5 Kapasitas Layanan per node.....	26
Gambar 3.6 Koneksi antar 2 Node	27
Gambar 4.1 Login router MR-3020	29
Gambar 4.2 Proses Upgrade firmware OpenWRT	30
Gambar 4.3 Tampilan awal firmware	30
Gambar 4.4 Partisi Flashdisk	31
Gambar 4.5 (a)Login root dan (b) proses pemberian perintah extroot	32
Gambar 4.6 Proses akhir tahapan extroot	33
Gambar 4.7 Urutan proses pemanggilan	35
Gambar 4.8 Pengujian call client-001 ke client-002.....	36
Gambar 4.9 Grafik pengaruh jarak terhadap troughput	41
Gambar 4.10 Grafik pengaruh jarak terhadap delay	41
Gambar 4.11 Grafik pengaruh jarak terhadap paket loss.....	42
Gambar 4.12 Pengaruh jumlah client terhadap throughput	43
Gambar 4.13 Pengaruh jumlah client terhadap delay	44
Gambar 4.14 Perbandingan Throughput pada native dan generic bridging.....	45
Gambar 4.15 Perbandingan Delay pada native dan generic bridging.....	46

Gambar 4.16 Throughput pada kapasitas panggilan antar node.....	47
Gambar 4.17 Delay pada kapasitas panggilan antar node.....	47
Gambar 4.18 Throughput pada kapasitas panggilan antar node	48
Gambar 4.19 Delay pada kapasitas panggilan antar node.....	49

DAFTAR TABEL

	Halaman
Tabel 2.1 Jenis Codec	13
Tabel 2.2 Kualitas Nilai Delay	14
Tabel 2.3 Kualitas Nilai Jitter	15
Tabel 2.4 Kualitas Nilai Paket Loss	15
Tabel 4.1 Hasil pengukuran jarak antara node dengan client	37
Tabel 4.2 Hasil pegujian kapasitas layanan per node dengan generic bridging....	38
Tabel 4.3 Hasil pegujian kapasitas layanan per node dengan native bridging.....	38
Tabel 4.4 Hasil pengukuran Kualitas koneksi antar node	39
Tabel 4.5 Data pengukuran kapasitas panggilan antar node.....	40

!!-- Halaman ini sengaja di kosongkan --!!

DAFTAR LAMPIRAN

Lampiran 7.1 Konfigurasi Penambahan Kapasitas Memori	55
Lampiran 7.2 Konfigurasi Native Bridging	55
Lampiran 7.3 Konfigurasi OLSR.....	64

!-- Halaman ini sengaja di kosongkan --!

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perkembangan telekomunikasi di dunia semakin pesat dan cepat. Diawali dengan adanya komunikasi konvensional dengan jaringan PSTN, dan berkembang menjadi komunikasi modern dengan jaringan seluler. Sejalan dengan perkembangan teknologi komunikasi yang semakin pintar, maka sarana komunikasi semakin meningkat jumlahnya dan beragam. Dengan meningkatnya jumlah sarana komunikasi, maka secara tidak langsung berakibat meningkat pula jumlah panggilan. Hal ini menjadikan kendala apabila besarnya jumlah panggilan tidak dapat dilayani oleh perangkat penerima panggilan komunikasi.

komunikasi VoIP(Voice Internet Protokol) adalah bentuk teknologi alternatif komunikasi pada saat ini. VoIP merupakan teknologi komunikasi suara jarak jauh yang dilewatkan pada media internet. Kelebihan komunikasi voip dapat menekan biaya menjadi rendah, karena jaringan bisa ikutkan dengan instalasi jaringan internet atau intranet yang sudah ada dan tranfer data dilewatkan melalui jaringan internet atau intranet.

Mobile Adhoc Network (MANet) sebagai jaringan wireless untuk tranfer data, yang memiliki node-node yang saling terhubung dalam kondisi bergerak. VoIP yang memanfaatkan MANet sebagai jaringan wireless untuk melewati data atau paket-paket suara, membutuhkan node-node sebagai sarana untuk memperluas area cakupan. Node/Server tersebut bisa dibangun dengan perangkat AP/Router dengan system embedded system OpenWRT. karena memberikan kelebihan mobilitas dan fleksibilitas dari sisi routing jaringan. Tetapi Server VoIP yang dibangun dengan menggunakan perangkat AP/Router ini, mempunyai keterbatasan. Salah satu keterbatasan tersebut adalah masalah kemampuan kapasitas dalam melayani jumlah panggilan oleh user/client yang terbatas atau sedikit. Sehingga hanya dapat diimplementasikan pada jaringan skala kecil. Keterbatasan tersebut disebabkan Semakin banyak panggilan pengguna yang masuk, sehingga

kerja server akan semakin berat dan berakibat server tidak dapat lagi mampu melayani panggilan. bila diimplementasikan pada jaringan skala besar dengan jumlah panggilan yang banyak akan terjadi penurunan pada kualitas layanan dikarenakan overhead.

Pada penelitian ini kami diusulkan suatu metode penanganan tranfer media data pada Server VoIP yaitu metode Native Bridging. Metode ini diharapkan mampu meningkatkan kapasitas jumlah panggilan yang dilayani oleh Server VoIP.

1.2 Rumusan Masalah

Dari latar belakang diatas didapat ditarik permasalahan, Voip dengan mengaplikasikan perangkat AP/Router sebagai VoIP Server, mempunyai kapasitas melayani jumlah panggilan yang terbatas atau sedikit, sehingga hanya cocok di aplikasikan pada jaringan skala kecil.

1.3 Batasan Masalah

Pada Penelitian ini, akan diberikan batasan masalah , yaitu :

1. Server VoIP menggunakan AP/Router TPLink -MR3020
2. VoIP codec yang digunakan jenis codec GSM
3. Client menggunakan Smartphone Android
4. Pengukuran jumlah panggilan dan kualitas voice
5. Skenario pengukuran secara real time, pada kondisi server dan client statis

1.4 Tujuan Penelitian

Tujuan penelitian ini, meningkatkan kapasitas panggilan VoIP yang mampu dilayani oleh Server VoIP berbasis embedded system dengan metode Native Bridging.

1.5 Manfaat Penelitian

1. Mengetahui peningkatan kapasitas jumlah maksimal panggilan yang dapat dilayani oleh server voip dengan metode Native Bridging.
2. Pada implementasi jaringan komunikasi, dapat ditentukan jumlah node. yang dibutuhkan untuk memenuhi jumlah pengguna pada suatu area.

1.6 Sistematika Penulisan

Sistematika penulisan pada penelitian ini terdiri dari beberapa bab, agar mudah dipahami, sebagai berikut :

BAB 1 : Pendahuluan

Pada bab pendahuluan berisi latar belakang, maksud dan tujuan penelitian, rumusan masalah, batasan masalah, manfaat dan metode penelitian yang digunakan, serta sistematika pembahasan.

BAB 2 : Tinjauan Pustaka

Pada bagian tinjauan pustaka berisikan teori pendukung yaitu tentang pengertian VoIP, MANet, protokol routing yang dipakai, jenis codec dan metode Native Bridging yang diujikan, serta aplikasi pendukung lainnya.

BAB 3 : Metoda Penelitian

Pada bab ini berisi tahapan tentang perencanaan penelitian meliputi penentuan hardware dan software yang akan digunakan, Instalasi software ke perangkat, konfigurasi protokol dan VoIP Codec, penentuan lokasi, skenario pengujian dan pengukuran untuk mendapatkan data.

BAB 4: Hasil dan Pembahasan

Pengimplementasian dari rancangan sistem dengan tahapan instalasi dan konfigurasi, pengujian dan pengukuran serta analisa data, dibahas pada bab ini. Yang meliputi data dan analisa panggilan sistem voip, kapasitas jumlah panggilan pernode, jarak jangkauan dan kapasitas panggilan dengan dua node

BAB 5 : Kesimpulan dan Saran

Pada bab ini berisikan kesimpulan dari hasil proses analisa .dan saran untuk menyempurnakan penelitian ini.

!-- Halaman ini sengaja di kosongkan --!!

BAB 2

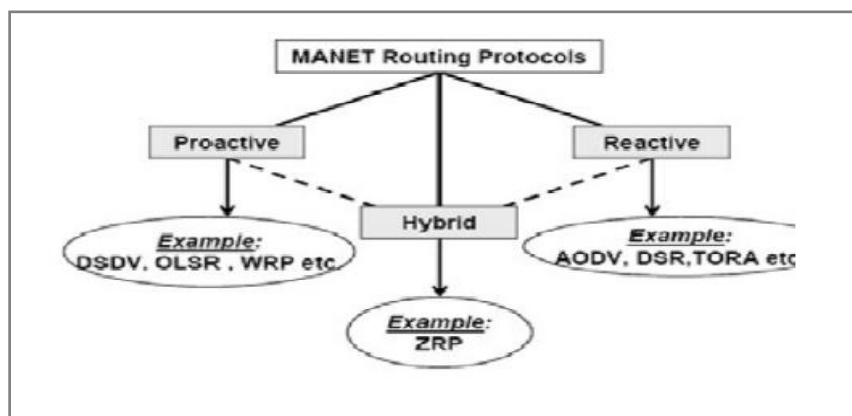
KAJIAN PUSTAKA

2.1 Manet Adhoc Network(MANET)

Mobile Adhoc Network atau yang lebih dikenal dengan MANET merupakan jaringan wireless yang memiliki node bergerak. Ciri khusus yang ada pada jaringan ini adalah setiap node yang dalam jaringan dapat melakukan proses routing dan pengiriman data. Node bertanggung jawab atas proses routing discovery dan routing maintenance untuk setiap jalur pengiriman data ke node destinasi. Hal ini dikarenakan setiap node yang berada dalam jaringan selalu bergerak. Adanya pergerakan yang random dari setiap node menyebabkan perubahan topologi pada jaringan MANET secara dynamic. Hal ini menyebabkan perubahan yang dynamic pada konektivitas antar node, yang saling bertukar data. Oleh karena itu diperlukan adanya routing protokol yang dapat mengcover kebutuhan jaringan untuk dapat memberikan jalur routing secara optimal (Samrat Ganguly, 2006).

2.2 Routing Protokol Pada Manet

Adanya perubahan yang dinamic pada konektifitas antar node pada jaringan MANet maka dibutuhkan routing protokol yang dapat menjaga koneksifitas yang handal pada jaringan MANet untuk proses pertukaran informasi.



Gambar .2.1 Klasifikasi protokol pada MANet

Pada gambar 2.1 menunjukkan klasifikasi routing protokol pada MANet yang dibedakan menjadi Protokol Reaktif, Proaktif dan Hybrid (Hrituparna Paul A, 2013).

2.2.1 Routing Protokol Reaktif

Jenis routing protokol ini, di kenal sebagai routing yang bekerja bila ada permintaan atau routing protokol yang bersifat reaktif. Pencarian rute terpendek untuk transmisi data baru akan dilakukan saat ada data yang akan dikirim. Akibat dari mekanisme tersebut adalah terjadinya delay yang sedikit lama saat akan melakukan pengiriman data. Protocol AODV, DSR dan TORA merupakan reaktif routing protokol.

2.2.2 Routing Protokol Proaktif

Routing protokol proaktif juga disebut sebagai *table driven* routing. Pada jenis routing protokol ini, tiap node diharuskan mempunyai tabel untuk menyimpan informasi daftar terbaru tujuan dan rute, distribusi tabel routing seluruh jaringan. Protokol ini selalu berusaha untuk menjaga konsisten dan memperbarui informasi routing pada setiap node (Hrituparna Paul A, 2013). Protokol routing proaktif menggunakan algoritma routing *link-state*, yang sering dipenuhi informasi dari node yang terhubung dengannya. Kelemahan utama proaktif routing protokol adalah bahwa semua node dalam jaringan selalu menjaga update tabel. Protokol OLSR, DSDV dan WRP adalah protokol routing proaktif (Saputra, 2013)

2.2.3 Routing Protokol Hybrid

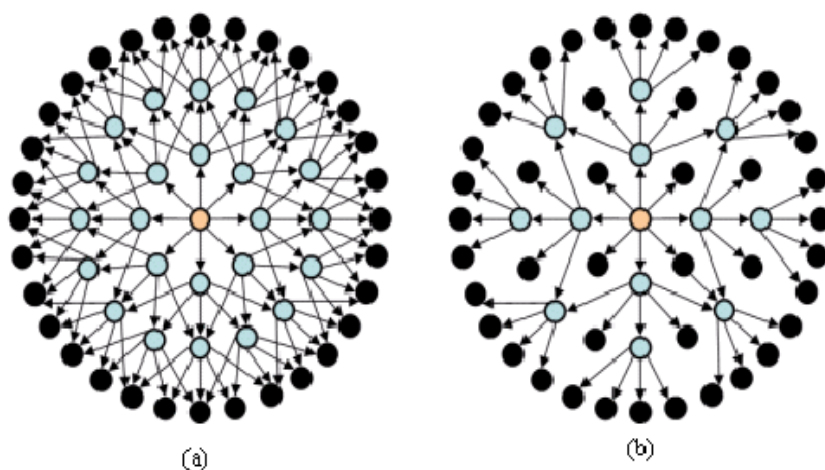
Routing protokol hybrid adalah routing protokol yang dibuat dengan mengambil keunggulan dari routing protokol reaktif dan proaktif, dengan kata lain menggabungkan keunggulan dari mekanisme kedua routing protokol tersebut. Dalam implementasinya, routing protokol jenis ini membutuhkan peralatan tambahan seperti GPS. Protokol ZRP adalah contoh dari routing protokol jenis ini.

2.2.4 Routing Protokol OLSR

Optimal Link State Routing (OLSR) Protokol adalah protokol routing proaktif dimana pada jenis routing ini jalur rute selalu tersedia bila diperlukan. OLSR merupakan versi optimasi dari protokol link state murni, di mana

perubahan topologi menyebabkan luapan dari informasi topologi ke semua host yang tersedia di jaringan, gambar 22.a. OLSR dapat mengoptimalkan reaktifitas perubahan topologi dengan mengurangi waktu maksimum interval untuk transmisi pesan control berkala. Selain itu, sebagaian OLSR terus mempertahankan rute untuk semua tujuan dalam jaringan, protocol ini bermanfaat bagi pola lalu lintas di mana subset node besar berkomunikasi dengan bagian node besar lainnya, dan pada saat pasangan berubah dari waktu ke waktu.

Protokol OLSR tidak cocok diaplikasikan pada transmisi paket data dengan keterlambatan lama dan bekerja dengan baik pada jaringan padat, dimana sebagian besar komunikasi terkonsentrasi antara sejumlah node besar. Sehingga terjadi adanya luapan informasi pada node utama. Untuk mengurangi jumlah *overhead* dalam jaringan yaitu, dengan menggunakan teknik *Multi Point Relays* (MPR). Tujuan utama dari MPR yaitu mengurangi luapan dengan cara memilih beberapa *node* untuk bertindak sebagai MPR, sehingga hanya *node-node* MPR saja yang dapat meneruskan paket kontrol yang diterima gambar 22.b Kinerja protokol OLSR yang paling efisien dalam jaringan padat. (Hriptuarna Paul A, 2013).



Gambar 2.2 Perbandingan Sistem broadcasting (a) Link state broadcast
(b) MPR broadcast

2.2.5 OLSR On Daemon

OLSR on Daemon (OLSRd) adalah bentuk implementasi dari OLSR routing protokol untuk MANet. Olsrd di rancang untuk memudahkan penggunaan

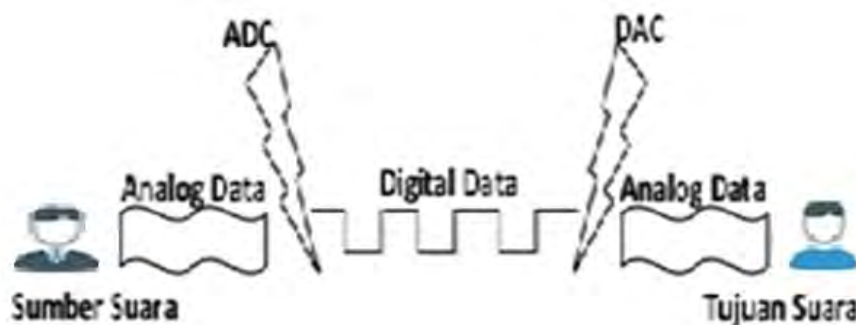
dan pemantauan dari penggunaan OLSR pada jaringan MANet serta penambahan plugin tertentu untuk mengaktifkan jaringan yang dibuat (Asriadi, 2014), (Hrituparna paul A , 2013). OLSRd dapat berjalan diberbagai platform seperti PC dekstop, single board computer, komputer berbasis ARM serta embedded device seperti router dan ponsel. Namun kebanyakan dari itu berjalan dengan sistem operasi yang berbasiskan GNU/Linux.

2.3 Voice Internet Protokol (VoIP)

VoIP (*Voice over Internet Protocol*) adalah teknologi yang mampu melewati trafik suara, *video* dan *text* yang berbentuk paket melalui jaringan IP. Jaringan IP adalah jaringan komunikasi data yang berbasis *packet-switch*, sehingga komunikasi telepon dapat dijalankan pada jaringan IP atau internet maupun intranet (Hrituparna paul A, 2013).

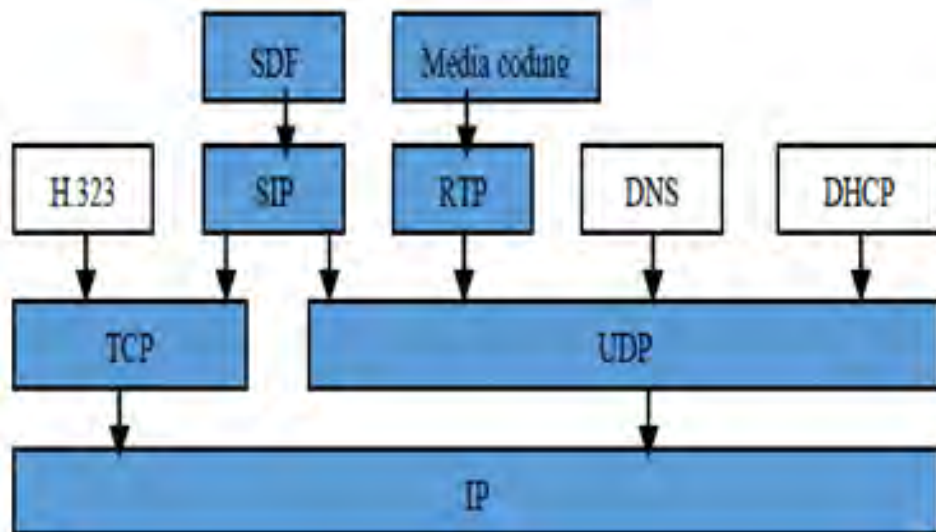
Pengiriman suara pada komunikasi VoIP dengan cara merubah suara *analog* menjadi data-data *digital*. Perubahan ini dilakukan dengan menggunakan *Analog to Digital Converter* (ADC/DAC) ditunjukkan pada Gambar 2.3. Data *digital* ini kemudian dikirimkan kepada penerima melalui jaringan Internet. Ketika data *digital* tersebut diterima, maka data *digital* tersebut akan di konversi kembali menjadi suara *analog* yang dapat didengarkan oleh penerima.

VoIP menjadi salah satu teknologi paling populer dan diprediksikan untuk menjadi komunikasi masa depan karena berbasis IP (Seto K, 2013). Teknologi VoIP menjanjikan biaya murah dalam pengembangan dan implementasinya serta fleksibilitas sistem baik itu untuk bisnis maupun individu.



Gambar 2.3 Cara Kerja VoIP

Selain ADC/DAC untuk merubah data analog ke data digital atau sebaliknya. Pada VoIP Juga bekerja beberapa protokol pendukung, berdasarkan fungsinya protokol pada VoIP dapat dibedakan menjadi 2 yaitu protokol persinyalan dan media transfer. Protokol-protokol VoIP tersebut ditunjukkan pada gambar 2.4 dibawah ini.



Gambar 2.4 Hirarki Protokol VoIP

2.3.1 Protokol Persinyalan .

Berfungsi untuk membangun, menjaga suatu sesi komunikasi yang sedang berlangsung dan memutus suatu koneksi. Berikut beberapa jenis protokol persinyalan.

a. **Session Initiation Protocol (SIP)**

Standarisasi protokol komunikasi pada teknologi VoIP adalah SIP (Session Initiation Protocol) dan H.323. Tetapi dalam perkembangannya SIP lebih banyak digunakan oleh pengembang teknologi VoIP (Asriadi, 2014). SIP merupakan sebuah session layer protokol yang digunakan untuk membentuk, memodifikasi, dan menterminasi sebuah sesi multimedia. Seperti layaknya HTTP, SIP merupakan client-server protokol yang menggunakan model transaksi request dan response. SIP juga telah banyak diadopsi oleh banyak software dan hardware dipasaran yang menjadikannya lebih mudah dalam implementasi (Istas P, 2013)

b. Inter Asterisk Exchange (IAX)

IAX merupakan signaling protokol yang dibuat sebagai alternatif dari SIP. Protokol tersebut bisa digunakan untuk komunikasi multimedia baik itu suara maupun video, namun terbatas hanya untuk komunikasi berbasis IP. Selain itu IAX adalah media dan signaling protocol point to point yang bekerja berbasiskan signal multiplexing dan multiplestreams melalui UDP diantara dua user pada port 4569.

Dibandingkan dengan SIP, IAX mempunyai beberapa kelebihan diantaranya konsumsi bandiwith yang lebih rendah, mendukung NAT transparan. Tetapi implementasi protokol ini ke peralatan-peralatan pendukung belum banyak dipasaran. Berbeda dengan SIP yang telah banyak diadopsi oleh banyak software dan hardware dipasaran yang menjadikannya lebih mudah dalam implementasi.

2.3.2 Protokol Media Tranfer

Berfungsi untuk mengatur komunikasi pada saat transfer data (baik *voice*, *video*, maupun data) secara *realtime*. Berikut adalah protokol-protokol media tranfer :

a) TPC/IP (*Transfer Control Protocol/Internet Protocol*)

TPC/IP merupakan protokol yang digunakan pada jaringan Internet. Dan terdiri dari dua bagian besar, yaitu TCP dan IP. Protokol TCP merupakan protokol yang connection-oriented yang artinya menjaga reliabilitas hubungan komunikasi end-to-end . Konsep dasar kerja TCP adalah mengirim dan menerima segment-segment informasi, dengan panjang data bervariasi pada suatu datagram internet.

TCP menjamin reliabilitas hubungan komunikasi karena melakukan perbaikan terhadap data yang rusak, hilang atau kesalahan kirim. Hal ini dilakukan dengan memberikan nomor urut pada setiap data yang dikirimkan dan membutuhkan sinyal jawaban positif dari penerima berupa sinyal ACK (Acknowledgment). Jika sinyal ACK ini tidak diterima pada interval pada waktu tertentu, maka data akan dikirimkan kembali.

Pada sisi penerima, nomor urut tadi berguna untuk mencegah kesalahan urutan data dan duplikasi data.

TCP juga memiliki mekanisme flow control dengan cara mencantumkan informasi dalam sinyal ACK mengenai batas jumlah oktet, data yang masih boleh ditransmisikan pada setiap segment yang diterima dengan sukses. Dalam hubungan dengan VoIP, TCP digunakan pada saat signaling, TCP digunakan untuk menjamin setup suatu call pada sesi signaling. TCP tidak digunakan dalam pengiriman data suara pada VoIP karena pada suatu komunikasi VoIP penanganan paket yang mengalami keterlambatan lebih penting daripada penanganan paket yang hilang (Dimas Lazuardi, 2008).

b) UDP (*User Datagram Protocol*)

UDP Merupakan salah satu protokol utama di atas IP dan merupakan protokol media transfer yang lebih sederhana, dibandingkan dengan TCP. UDP digunakan untuk situasi yang tidak mementingkan mekanisme reliabilitas. Artinya pada protokol UDP ini komunikasi akan tetap berlangsung tanpa memperdulikan koneksi antara sumber dan tujuan. Protokol UDP juga tidak melakukan perbaikan terhadap paket yang rusak atau hilang pada saat pengiriman paket suara berlangsung. Header UDP hanya berisi empat field yaitu source port, destination port, length dan UDP checksum yang fungsinya hampir sama dengan TCP, namun fasilitas checksum audio streaming yang secara terus menerus pada UDP bersifat opsional.

UDP pada VoIP digunakan untuk mengirimkan audio streaming yang secara terus menerus. Protokol ini lebih mementingkan kecepatan pengiriman paket data agar tiba di tujuan tanpa memperhatikan adanya paket yang hilang walaupun mencapai 50% dari jumlah paket yang dikirimkan. Karena UDP mampu mengirimkan data streaming dengan cepat, sehingga dalam teknologi VoIP, UDP merupakan salah satu protokol penting yang digunakan sebagai header pada pengiriman data.

c) RTP (*Real Time Protocol*)

RTP adalah protokol yang didesain untuk membawa paket audio atau video melalui jaringan berbasis IP. Paket pada RTP memuat potongan paket-paket audio ataupun video yang ditransmisikan dengan bantuan protokol UDP. Karena bekerja pada UDP maka jika terdapat paket RTP yang tidak diterima dengan baik atau hilang saat transmisi, maka RTP tidak akan melakukan pengiriman paket ulang.

d) SRTP (*Secure Real Time Protokol*)

Protokol SRTP sebagai penyedia fitur enkripsi pada profil RTP. Fitur enkripsi ini diperuntukkan sebagai sistem keamanan data, dengan otentikasi dan integritas pesan serta perlindungan terhadap playback data RTP dalam aplikasi unicast dan multicast. Dengan metode algoritma AES (*Advanced Encryption Standard*) untuk mengenkripsi data dalam proses pengiriman.

Pada aplikasi, SRTP terdiri dari 3 mode, yaitu Segmented AES di f8-mode, Integer Counter dan null cipher. Pada Mode Null Cipher, pengiriman data tidak terlindungi oleh adanya enkripsi. SRTP pada kenyataannya hanya mengenkripsi payload (audio dan video) untuk kerahasiaan.

2.3.3 Coder dan Decoder (Codec) VoIP

Codec merupakan sebuah proses mengubah data suara (analog) ke dalam bentuk data digital yang selanjutnya dimampatkan (Kompresi), kemudian ditransmisikan dan dikembalikan lagi ke bentuk sinyal suara ketika sampai ke tujuan secara *realtime* melalui *jaringan Internet Protokol* (Asriadi, 2014). Sehingga peranan codec pada komunikasi VoIP sangat penting sekali. Karena besar kecilnya hasil proses kompresi data oleh codec dapat menghemat bandwidth dan mempengaruhi kualitas suara pada komunikasi VoIP.

Codec tersedia dalam bentuk *open source* dan *non-open source*. Tabel 2.1 menunjukkan daftar dari beberapa jenis codec.

Tabel 2.1 Jenis Codec

Codec	Bitrate (kb/s)	Frame (ms)	Bits per frame	Algorithmic delay ^a (ms)	Codec delay ^b (ms)	Compression type	Complexity (MIPS) ^c	MOS
<i>Narrowband codecs</i>								
G.711	64	0.125	8	0.125	0.25	PCM	≪1	4.1 ^d
G.723.1	6.3	30	189	37.5	67.5	MP-MLQ	≪18	3.8
G.723.1	5.3	30	159	37.5	67.5	ACELP	≪18	3.6
G.726	16	0.125	2	0.125	0.25	ADPCM	≈1	-
G.726	24	0.125	3	0.125	0.25	ADPCM	≈1	3.5
G.726	32	0.125	4	0.125	0.25	ADPCM	≈1	4.1
G.728	16	0.625	10	0.625	1.25	LD-CELP	≈30	3.61
G.729	8	10	80	15	25	CS-ACELP	≪20	3.92
G.729A	8	10	80	15	25	CS-ACELP	≪11	3.7
G.729D	6.4	10	64	15	25	CS-ACELP	<20	3.8
G.729E	11.8	10	118	15	25	CS-ACELP LPC	<30	4
GSM-FR	13	20	260	20	40	RPE-LTP	≈4.5	3.6
GSM-HR	5.6	20	112	24.4	44.4	VSELP	≈30	3.5
GSM-EFR	12.2	20	244	20	40	ACELP	≈20	4.1
AMR-NB	4.75–12.2	20	95–244	25	45	ACELP	15–20	3.5–4.1
iLBC	13.33	30	400	40	60	LPC	18	3.8
iLBC	15.2	20	304	25	40	LPC	15	3.9
Speex (NB)	2.15–24.6	20	43–492	30	50	CELP	8–25	2.8–4.2
BV16	16	5	80	5	10	TSNFC	12	4
<i>Broadband codecs</i>								
G.722	48, 56, 64	0.0625	3–4	1.5	1.5625	SB-ADPCM	5	~4.1
G.722.1	24, 32	20	480, 640	40	60	MLT	<15	~4
AMR-WB (G.722.2)	6.6–23.85	20	132–477	25	45	ACELP	≈38	Various
Speex (WB)	4–44.2	20	80–884	34	50	CELP	8–25	Various
iSAC	Variable	Adaptive	Adaptive-variable	Frame + 3 ms	Adaptive	Transform	6–10	Various ^e
	10–32	30–60 ms			63–123	coding		
BV32	32	5	160	5	10	TSNFC	17.5	~4.1

Jenis codec yang banyak digunakan pada komunikasi voip adalah codec GSM (Global System Mobile). Codec ini merupakan salah satu jenis codec VoIP yang mempunyai kinerja sangat baik untuk kompresi audio (OpenWRT, 2012).

Codec GSM merupakan jenis pengkodean suara yang melakukan pengompresian paket suara dengan teknik RPE-LTP (*Regular Pulsa Excitation Long Term prediction*). Codec ini menggunakan sampel sebanyak 16.000 kali/detik dengan bit rate 8 bit/detik sehingga menghasilkan laju rate sebesar 13.000 bit/detik. RPE-LTP mengkonversi sinyal analog ke bentuk digital dengan melakukan sampling sinyal analog tersebut 8000 kali per detik dan dikodekan dalam angka. Jarak antar sampel adalah 20,000 μ s. Sinyal tersampel lalu

dikonversikan ke bentuk diskrit. Sinyal diskrit ini direpresentasikan dengan kode yang disesuaikan dengan amplitude dari sinyal sampel. Format RPE-LTP menggunakan 260 bit untuk pengkodeannya. Laju transmisi diperoleh dengan mengalikan 8000 sampel per detik dengan 260 bit per sampel, menghasilkan 13.000 bit per detik. Bit rate 13 Kbps ini merupakan standar transmisi untuk suatu kanal telepon digital.

2.3.4 *Quality Of Service (QOS)*

Suatu metode pengukuran kemampuan dalam menyediakan tingkat layanan untuk transmisi data pada suatu jaringan. Adapun parameter QOS adalah :

a) *Throughput*

Throughput menunjukkan jumlah *bit* yang diterima dengan sukses per detik melalui sebuah sistem atau media komunikasi dalam selang waktu tertentu yang pada umumnya dilihat dalam satuan *bits/sec*

b) *Delay*

Delay menunjukkan waktu tunda yang terjadi pada suatu data ketika ditransmisikan dari *transmitter* menuju *receiver*. Waktu tunda merupakan suatu permasalahan yang harus diperhitungkan karena kualitas suara bagus tidaknya tergantung dari waktu tunda. Besarnya waktu tunda maksimum yang direkomendasikan oleh ITU-T G.114 untuk aplikasi suara adalah 150 ms, sedangkan waktu tunda maksimum dengan kualitas suara yang masih dapat diterima pengguna adalah 250 ms.

Tabel 2.2 Kualitas Nilai Delay

Nilai Delay	Kualitas
0 - 150 ms	Baik
150 - 250 ms	Cukup
> 250 ms	Buruk

c) *Jitter*

Jitter merupakan variasi delay yang terjadi karena adanya selisih waktu antar kedatangan paket yang di penerima. Untuk mengatasi jitter, paket data yang datang dikumpulkan terlebih dulu dalam buffer selama waktu

yang telah ditentukan sampai paket dapat diterima keseluruhan pada sisi penerima, dengan urutan yang benar. Adanya buffer tersebut akan mempengaruhi waktu tunda total sistem, akibat adanya tambahan proses untuk mengompensasi terjadinya jitter. Pada tabel 2.3 menunjukkan, jitter berkategori baik jika mempunyai 0 – 20 ms dan masih dalam batas toleransi apabila bernilai 20 – 50 ms, sedangkan kategori buruk jika nilai jitter lebih dari 50 ms

Tabel 2.3 Kualitas Nilai Jitter

Nilai Jitter	Kualitas
0-20 ms	Baik
20-50 ms	Cukup
>50 ms	Buruk

d) Paket Loss

Packet Loss adalah hilangnya paket data yang dikirim ketika terjadi Peak load dan congestion (kemacetan transmisi paket akibat padatnya traffic yang harus dilayani) dalam batas waktu tertentu, maka frame (gabungan data payload dan header yang di transmisikan) suara akan dibuang, sebagaimana perlakuan terhadap frame data lainnya pada jaringan berbasis IP, sehingga ada kemungkinan terjadinya hilang paket pada saat transmisi. Tingkat paket hilang ini tentunya akan mempengaruhi kualitas layanan. Pada tabel 2.4, menunjukkan packet loss berkualitas baik apabila 0 – 1,5%, dan tidak dapat di terima apabila > 1,5% dalam aplikasi suara.

Tabel 2.4 Kualitas Nilai Paket Loss

Packet Loss	Kualitas
0-0,5 %	Sangat baik
0,5-1,5 %	Baik
>1,5 %	Buruk

2.4 Operating Sistem untuk Router Wireless

OpenWRT merupakan salah satu distribusi linux yang didesain untuk perangkat *embedded*. Salah satu penerapan sistem operasi OpenWRT adalah pada wireless router.

OpenWRT diciptakan semula untuk mendukung *wireless router* Linksys WRT54G, tapi perkembangan OpenWRT selanjutnya dapat digunakan pada beberapa manufaktur wireless router yang lain seperti TP-Link, D-Link, ASUS, DELL dan lain-lain. OpenWRT menyediakan *file system* dengan modul-modul dari aplikasi terpisah yang dapat diimplementasikan pada OpenVoice (Asriadi,2014). *Modul-modul* tersebut di antaranya uClibc, busybox, shell interpreter, abstraksi perangkat keras dan juga paket manager Aplikasi Voip

2.4.1 Sistem pada Server VoIP

Asterisk adalah framework yang bersifat open source yang digunakan untuk membangun sebuah jaringan komunikasi. Asterisk mencakup banyak fitur yang tersedia dalam sistem PBX seperti voice mail, teleconference, VoIP, dan distribusi panggilan otomatis.

Asterisk mendukung berbagai fitur multimedia seperti Voice over IP protokol, dengan menggunakan protokol Session Initiation Protocol (SIP), Media Gateway Control Protocol (MGCP), dan H.323. Asterisk dapat berperan baik sebagai registra atau sebagai gateway antara VoIP dan PSTN. Kelebihan asterisk dengan ukuran sistem yang kecil, sehingga aplikasi ini memungkinkan aplikasi ini berjalan pada embedded system seperti OpenVoice (OpenWRT, 2014). dan menjadikan router biasa menjadi sebuah VoIP Server

Asterisk adalah salah satu paket modul yang diinstal pada OpenWRT dan akan berada pada bagian User Program. Paket-paket yang membutuhkan akses ke hardware akan melewati Linux Kernel dan kemudian akan diteruskan ke hardware tujuan. Asterisk itu sendiri memberikan kesederhanaan bagi pengguna untuk meningkatkan layanan telepon sendiri dengan kustomisasi yang fleksibel oleh pengguna.

2.4.2 TP Link MR-3020

Perangkat TP Link MR3020 adalah sejenis Router/Akses Point wireless. Yang mempunyai spesifikasi sebagai berikut, berdimensi 2.9 x 2.6 x 0.9 inci (74 x 67 x22 mm), karena ukurannya yang sangat kecil sehingga mudah untuk dibawa. Berdaya 5VDC/1.0A konsumsi listrik sangat kecil. Mempunyai frekuensi 2.4-2.4835GHz. Didukung System-On-Chip(SoC) Atheros AR9331, CPU Speed 400MHz, RAM 32 MB, Flash Memory 4 MB Wireless Chip Atheros AR9331 1x1:2, perangkat ini selain mempunyai spesifikasi diatas, yang lebih penting type MR3020 ini dapat ditanam aplikasi firmware OpenWRT, asteriks dan dapat ditambahkan external storage. Sehingga perangkat router jenis ini sangat cocok dan sesuai kebutuhan, untuk dijadikan sebagai Server VoIP pada penelitian ini (Asriadi, 2014).

Spesifikasi lengkap router MR-3020 yang akan dipakai dalam penelitian ini dapat dilihat pada Tabel 2.5.

Tabel 2.5 Spesifikasi Router MR-3020

No	Spesifikasi	Ket.
1	Arsitektur	MIPS 24Kc V7.4
2	Vendor	Atheros
3	Bootloader	U-boot 1.1.3
4	System-On-Chip	Atheros AR9331
5	CPU Speed	400MHz
6	RAM Memory	32 MB SDRAM
7	Flash Memory	4 MB
8	Flash Chip	Windbond W9425G6JH
9	Wired Network	2x Ethernet 100 Mbps (switched)
10	Wireless Chip	Atheros AR9331 1x1:2
11	Wireless antennas	2x printed on-board
12	USB	1 x USB 2.0 host
13	Serial Port	Yes (TTL pins)

2.4.3 Sistem pada VoIP Client

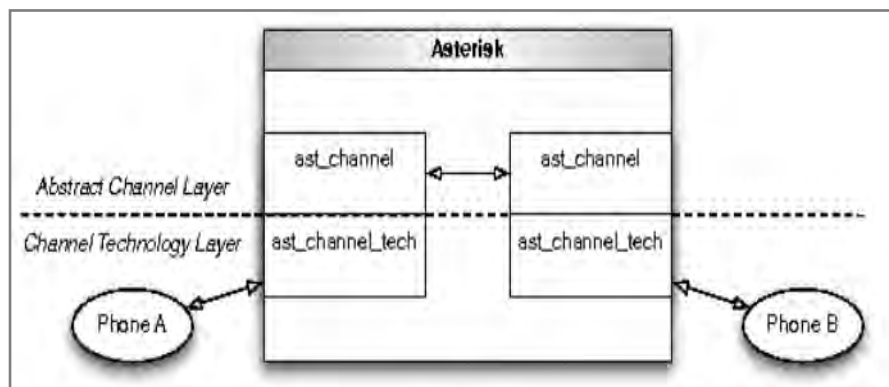
Pada sisi client, perangkat komunikasi yang digunakan berupa smartphone dengan sistem operasi Android, bisa juga sebuah seperangkat PC multimedia dengan menggunakan sistem operasi windows maupun sistem operasi opensource.

Softphone adalah aplikasi komunikasi voip pada client yang menggunakan smartphone. Aplikasi softphone memiliki jenis yang beragam baik dari kemampuan dan lisensi. Saat ini banyak softphone yang disebarakan dengan lisensi gratis, bahkan ada yang menyediakan software lisensi gratis sekaligus untuk layanan jaringan VoIP-nya. Aplikasi softphone tersebut diantaranya adalah Zoiper. Softphone Zoiper merupakan aplikasi yang dapat digunakan untuk komunikasi suara melalui VoIP.

Sedangkan untuk PC yang menggunakan sistem operasi windows atau opensource, aplikasi yang dapat digunakan antara lain X-Lite, IAX-Lite, dan MyPhone. X-Lite merupakan aplikasi untuk VoIP yang berjalan melalui protokol SIP. X-Lite juga bisa digunakana untuk saling berkirim suara, text dan video.

2.5 Bridging

Pada umumnya proses komunikasi pada voip menggunakan model saluran *Generic Bridging*. Saluran Generic Bridging adalah metode penanganan tranfer media suara pada voip, yang menggunakan saluran inti server (asterisk) secara bersama-sama sebagai media tranfer untuk tujuan melewati signal call, dan data suara antar client.

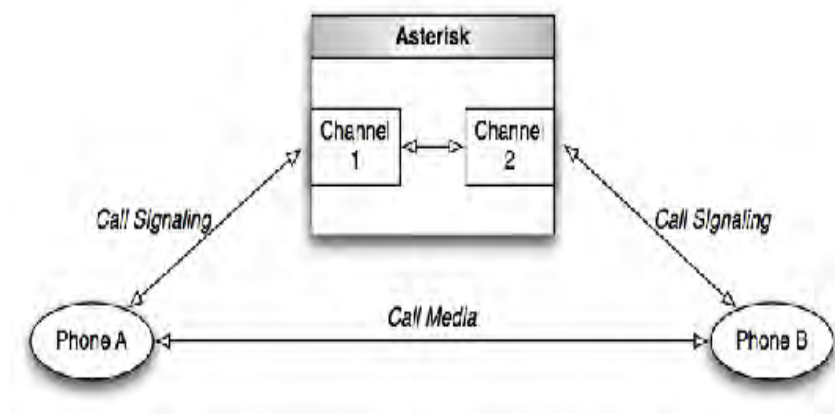


Gambar 2.5 Saluran Bridging

Dapat dikatakan semua proses komunikasi yang terjadi, melewati inti server termasuk proses transcoding codec. Dapat dilihat pada Gambar 2.5 di mana dua kaki panggilan diwakili oleh dua saluran ketika terhubung ke saluran inti server (Asterisk). Aliran paket media selain video dan teks dalam panggilan yang melewati inti server, yang paling sering yaitu audio (Anonymous Asterisk, 2014).

Native Bridging adalah metode penanganan tranfer media suara pada voip dengan cara bypass tanpa melalui inti server. Gambar 2.6, menunjukkan dimana setelah komunikasi(call signal) tersambung antar client (Phone A dan Phone B) pada inti server, selanjutnya paket data suara tidak dilewatkan ke inti server (asterisk) lagi untuk proses transcoding. Tetapi paket suara (Call media) dari kedua client langsung dipertukarkan antar client (endpoint).

Pada native bridging server tidak melakukan transcoding codec pada inti asterisk, maka sebagai syarat syarat codec yang dipakai harus sejenis. Dengan tidak melewati media suara pada inti server untuk proses transcoding codec, akan menyingkat waktu proses tranfer media. sehingga dapat menaikkan jumlah client yang mampu dilayani secara bersamaan. Akibat ketika kedua client tidak suport codec yang sama, maka komunikasi tidak akan tersambung (Effan Najwaini, 2014).



Gambar 2.6 Skema Native Bridging

2.6 Aplikasi Pengukuran

Untuk pengukuran dan menganalisa proses kerja jaringan, bisa digunakan beberapa software analizer jaringan, diantara yaitu: *Wireshark* merupakan perangkat lunak yang spesifik untuk melakukan analisa paket data pada jaringan secara real time dan menampilkan hasil analisa paket data tersebut dalam format yang dipahami oleh pengguna. Wireshark dapat melakukan paket filtering, paket color coding, dan fitur-fitur lain yang dapat mengizinkan untuk melihat detail network traffic dan inspeksi paket data secara individu.

Wireshark dapat menganalisis paket data secara real time. Artinya, aplikasi wireshark akan mengawasi semua paket data yang keluar masuk melalui antarmuka yang telah ditentukan dan selanjutnya akan menampilkan hasil paket datanya. Wireshark dapat melakukan analisis terhadap beberapa protokol paket data jaringan. Termasuk mendeteksi, menganalisa paket-paket data yang lewat pada komunikasi VoIP

2.7 Penelitian Terkait

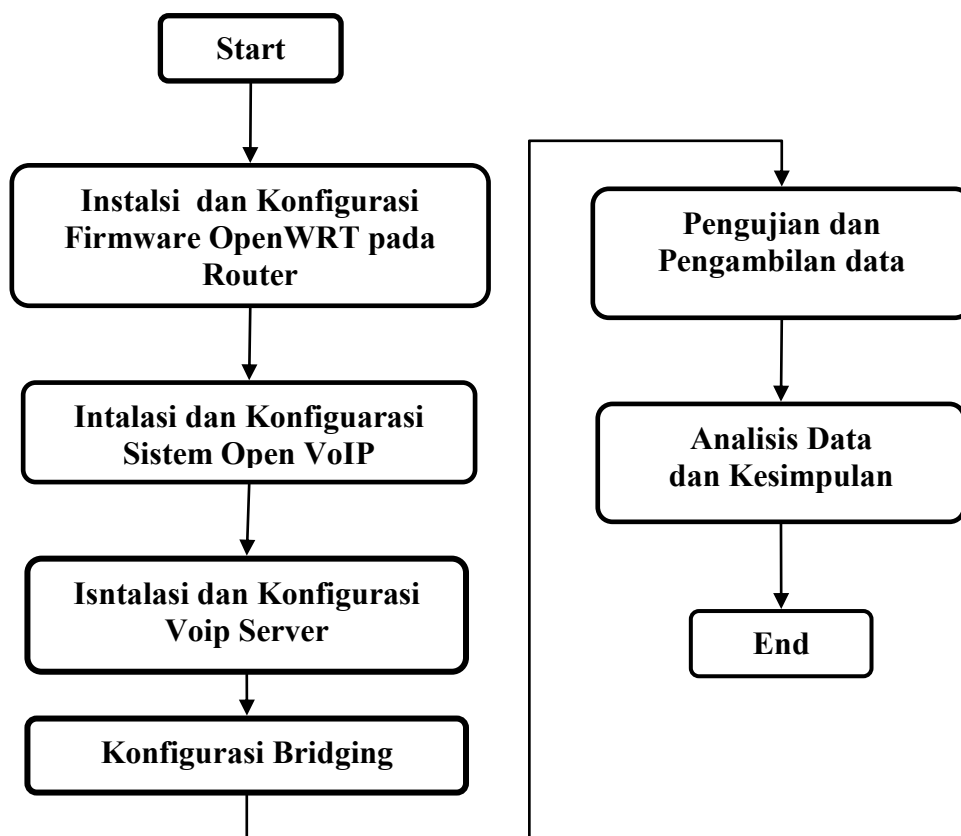
Ada banyak penelitian dengan riset tentang pengembangan komunikasi voip, dengan tujuan untuk menyediakan komunikasi dengan sarana dan biaya yang murah, baik itu untuk internal atau dimanfaatkan untuk tempat atau daerah yang belum terjangkau oleh sarana komunikasi, seperti daerah terpencil dan daerah bekas bencana. Kebanyakan penelitian dan project tersebut menggunakan perangkat dengan harga yang terjangkau oleh masyarakat dan mudah di dapat. Project tersebut seperti: desain dan prototype wifi bts untuk daerah terpencil(Asriadi, 2014,) penerapan biaya rendah pada komunikasi VoIP (istas P.2013), beberapa riset kinerja dan kualitas elemen pendukung komunikasi voip, seperti analisa kinerja voip server pada AP Wireless(Effan N, 2014), perbandingan routing protokol voip (hriptuparna Paul, 2013), perbandingan protokol pada MANet(samrat G,2006) analisa kualitas kanal voip (Risky Agri S, 2010). Penelitian –penelitian tersebut diatas ditujukan untuk pengembangan dan penerapan komunikasi Voip. Sebagai teknologi alternatif yang murah untuk komunikasi.

BAB 3

METODOLOGI PENELITIAN

3.1 Rancangan Penelitian

Dalam perancangan penelitian ini, akan dilakukan tahapan-tahapan untuk membangun sistem yang akan buat, tahapan tersebut terlihat pada gambar 3.1. Diagram Flowchart Sistem.



Gambar 3.1 Digram Flowchat Sistem

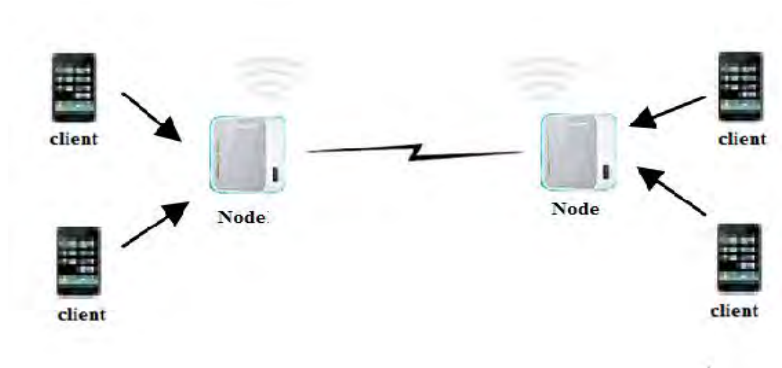
3.2 Gambaran Umum Sistem

Pada desain rancangan sistem Server VoIP yang dibuat dengan menggunakan firmware yang berbasis OpenWRT Barrier Breaker1.4 sebagai

frameworknya, diembeddedkan pada perangkat router wireless TP Link MR-3020.

Sebagai softswitch untuk membangun SIP Server (Server VoIP) akan digunakan aplikasi open source asterisk, yang berfungsi untuk melayani suara pada jaringan wireless voip. Di mana akan mengkonfigurasi asterisk pada mode native bridging untuk meningkatkan kapasitas layanan panggilan pada Server VoIP .

Pada sisi client, menggunakan smartphone sebagai perangkat komunikasi, dengan sistem operasi android. Dan telah terinstal dengan aplikasi softphone Zoiper sebagai media untuk berkomunikasi di client, yang terhubung secara wireless pada frekuensi ISM band 2,4 GHz kepada node-node yang ada. Arsitektur jaringan pada saat pengimplementasian prototype ini pada jalur akses dapat diilustrasikan seperti pada gambar 3.2



Gambar 3.2 Arsitektur Jaringan VoiP

3.2.1 Tahapan Instalasi Sistem OpenWRT pada Router

Tahapan awal untuk membangun sebuah server voip, adalah melakukan instalasi sebuah firmware OpenWRT Barrier Breaker 1.4 yang suport dengan perangkat AP/router TP Link MR3020 yang akan dijadikan sebagai VoIP Server, di mana firmware tersebut akan terembedded pada perangkat. Firmware pada embedded sistem berbasis OpenWRT sebagai frameworknya, asterisk sebagai softswitch dan digunakan dalam membuat SIP server untuk layanan suara serta prosody untuk layanan pesan. Digunakan firmware khusus ini bertujuan untuk

mendapatkan firmware yang sesuai dengan kebutuhan aplikasi. Karena dalam firmware tersebut telah built-in dengan modul-modul yang diinginkan seperti asterisk, prosody, routing protokol serta mendukung beberapa jenis codec, sehingga dapat dipakai untuk komunikasi pesan dan suara serta dapat digunakan untuk meningkatkan quality of service dari sistem.

3.2.2 Tahapan Instalasi Open VoIP Sistem

Pada tahapan ini akan dilakukan beberapa instalasi dan konfigurasi adalah :

a. Konfigurasi Extroot

Keterbatasan spesifikasi hardware terutama pada internal storage yang minim kapasitas, bisa kita atasi dengan cara extroot. Dimana konfigurasi extroot ini merupakan salah satu cara untuk menambah internal storage pada AP/router yang sudah embedded openwrt, dengan menggunakan perangkat external storage berupa flashdisk.

b. Konfigurasi Server VoIP

Pada tahapan ini akan dilakukan konfigurasi asterisk yang akan menjadi SIP server dalam melayani pengguna yang terhubung dengan system ini.

c. Konfigurasi Native Bridging

Pada tahapan ini, akan dilakukan konfigurasi pada file sip.conf dan extension.conf. Konfigurasi ini menambahkan beberapa variabel khusus dan merubah nilai standart. sehingga akan mengaktifkan mode bridging menjadi Native Bridging.

3.3 Skenario Pengujian

Pada tahap ini akan dilakukan pengujian dan pengukuran terhadap kemampuan Server VoIP, yang menggunakan perangkat Router TP Link MR3020 yang telah diembeddedkan firmware, routing protokol OLSR, dan codec yang digunakan adalah jenis codec GSM serta mengkonfigurasi saluran bridging menjadi mode native bridging untuk mendukung penerapan VoIP dalam sistem ini, menjadi sebuah Server VoIP .

Pengujian ini dibagi dalam beberapa skenario. Hal ini dilakukan agar mendapatkan hasil pengukuran yang mendekati pada saat benar-benar

diimplementasikan. Berikut penjelasan mengenai skenario serta kondisi yang diterapkan saat pengukuran dilakukan :

1. Kondisi smarphone yang dipakai tidak dalam keadaan terpasang simcard operator, untuk mensimulasikan tidak adanya layanan operator yang dipakai. Begitu pula dengan kondisi jaringan yang dipakai, tidak ada sama sekali layanan internet yang dipakai sehingga sistemnya benar-benar bekerja dalam keadaan offline
2. Semua pengujian dan pengukuran dilakukan secara real time, tidak menggunakan aplikasi simulator, artinya pengujian dilakukan dengan melakukan pemanggilan secara langsung
3. Lokasi yang digunakan dalam pengukuran adalah tanah lapang untuk mengetahui performance dari sistem pada saat keadaan sinyal bebas halangan
4. Skenario pengujian dalam kondisi statis : dimana semua node dan Client yang ada di dalam sistem dalam keadaan diam dan tak bergerak.

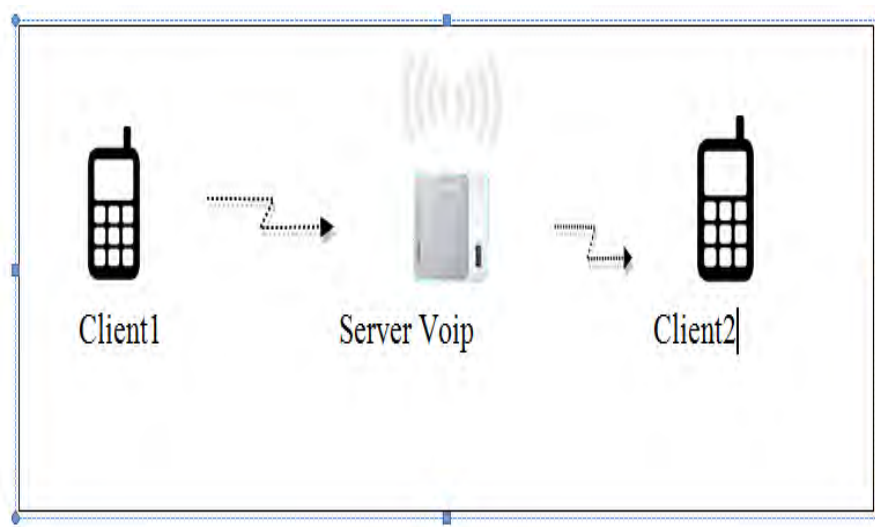
3.3.1 Pengujian Panggilan pada Sistem

Pengujian ini dilakukan untuk menguji kerja sistem yang sudah dibuat. Pengujian ini dilakukan dengan dua cara : pengujian pertama melakukan panggilan dari client-1 kepada client-2 tanpa jarak dan pengujian kedua dengan menentukan jarak panggilan.

a. Pengujian Sistem Server Voip

Pengujian dilakukan untuk mengetahui bahwa sistem dan konfigurasi yang dibuat pada server voip berjalan sesuai dengan rancangan. Pada gambar 3.3 menunjukkan pengujian dilakukan dengan cara melakukan panggilan dari client-1 kepada client-2 tanpa jarak. Pengujian sistem dinyatakan berhasil apabila client-1 dapat memanggil dan client-2 dapat menerima panggilan, serta dapat melakukan percakapan. Pengujian sistem tidak berhasil apabila :

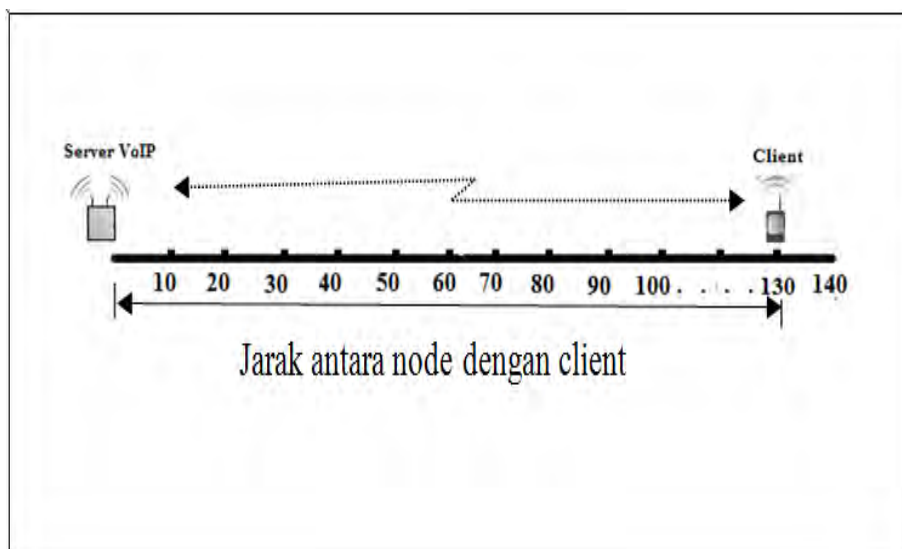
1. Signal panggilan terputus apabila client-2 menerima panggilan.
2. Salah satu client tidak bisa menerima panggilan.
3. Salah satu client tidak terdengar suara percakapan.



Gambar 3.3 Call antar client

b. Pengujian dengan jarak panggilan

Pada pengujian ini akan dilakukan uji pemanggilan dari client1 ke client2. Jarak awal antara client1 dengan client2 adalah 10 meter, untuk jarak panggilan selanjutnya kelipatan 10 meter seperti terlihat pada gambar 3.4. Pengujian ini dilakukan untuk mencari kemampuan jarak terjauh yang dapat di cover node. Sehingga dapat diperkirakan jarak penempatan tiap node pada area implementasi.



Gambar 3.4 Jarak panggilan antara node dengan client

3.3.2 Pengujian Kapasitas Maksimum Panggilan Per Node

Pengujian ini berguna untuk mengetahui kapasitas layanan maksimum server VoIP sebagai node, sehingga dapat memperkirakan jumlah node yang harus dipasang pada saat implementasi nanti diilustrasikan pada gambar 3.5. Pengujian dilakukan dengan panggilan masuk sebanyak mungkin secara simultan, dengan pengaturan waktu tertentu tiap panggilan.

Cara pemanggilan dilakukan dengan berpasangan antar client. Pemanggilan pertama dan berikutnya tetap pada kondisi terkoneksi, dan melakukan percakapan sampai pemanggilan terakhir terhubung, hingga server voip tidak dapat melayani panggilan atau adanya panggilan sebelumnya yang terputus.

Tiap kali ada panggilan masuk akan dilakukan proses pengambilan data dan dihitung total panggilan yang berhasil. Hasil dari pengujian ini merupakan kemampuan real time dari server voip, dengan mengaplikasikan metode native bridging. Data hasil pengujian akan dibandingkan dengan data hasil pengujian kapasitas panggilan server voip tidak menggunakan metode native bridging.



Gambar 3.5 Kapasitas Layanan per node

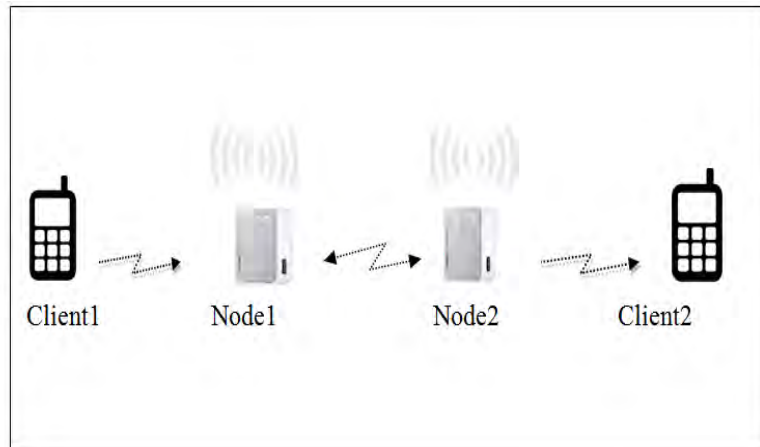
3.3.3 Pengujian Kualitas Koneksi dan Kapasitas Panggilan Antar Node

Pengujian ini dilakukan dengan cara menempatkan dua node atau lebih, di mana kondisi antar node tersebut saling terkoneksi, dan akan dilakukan pemanggilan dari client yang terkoneksi dengan node yang berbeda.

a) Pengujian kualitas koneksi antar node

Dari pengujian ini akan di ketahui sistem bekerja dengan baik pada sistem jaringan, apabila antar client bisa melakukan panggilan dan menerima

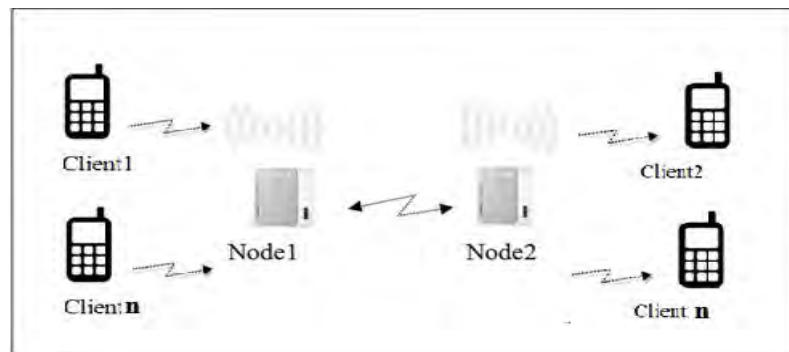
(saling terhubung) serta dapat melakukan percakapan. Berikut gambar 3.6, model dari pengujian antar node



Gambar 3.6 Pengujian koneksi antar 2 Node

b) Pengujian kapasitas maksimal layanan antar node.

Pada gambar 3.7 menunjukkan pengujian dan pengukuran kapasitas panggilan yang mampu di layani oleh node server pada kondisi terkoneksi dengan node lain. Cara pengujian ini hampir sama dengan pengujian kapasitas per node. Hanya yang koneksi masing-masing client ke node-1 atau node-2 saja yang membedakan.



Gambar 3.7 Pengujian Kapasitas Panggilan antar 2 Node

BAB 4

HASIL PENELITIAN DAN PEMBAHASAN

4.1 Implementasi Rancangan

Pada bab ini dilakukan implementasi rancangan sistem, dengan tahapan instalasi dan konfigurasi sistem untuk membangun sebuah Server VoIP.

4.1.1 Instalasi Sistem OpenWRT pada Router

Setelah ditentukan perangkat, tahapan awal implementasi rancangan adalah menginstal firmware OpenWRT Barrier Breaker 1.4 sebagai framework komunikasi berbasis embedded sistem yang di implementasikan pada Wireless VoIP.

Pada tahapan ini akan dilakukan instalasi firmware ke dalam device yang telah ditentukan. Langkah-langkah instalasinya sebagai berikut :

- a) Siapkan file firmware OpenWRTt Barrier Breaker 1.4, file ini bisa di download bebas di internet.
- b) Masuk kedalam halaman administrator dari device yang kita gunakan melalui browser. Pada kasus ini halaman administrator dapat diakses di <http://192.168.0.254>, setelah kita masukkan username dan passwordnya maka kita akan mendapati tampilan seperti pada gambar 4.1



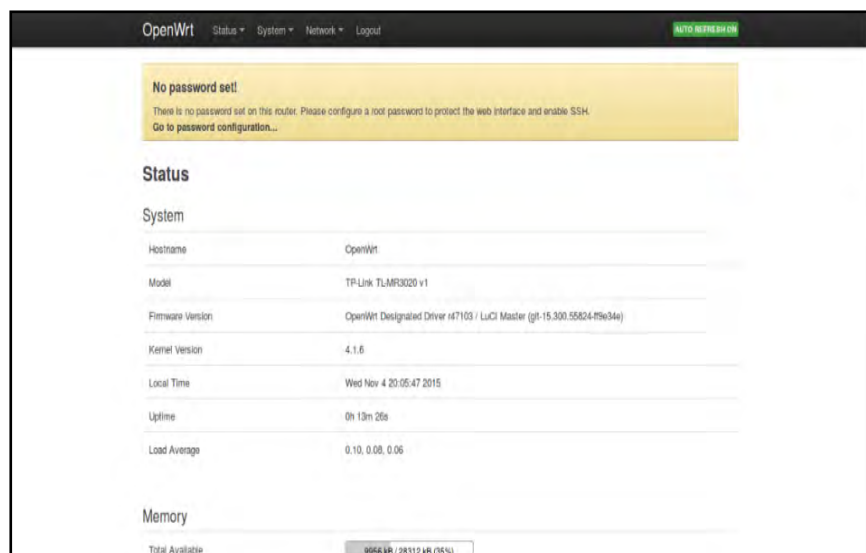
Gambar 4.1 Login router MR-3020

- c) Pilih menu sistem tools kemudian pilih firmware upgrade lihat gambar 4.2 dibawah. Pilihlah firmware OpenWRT Barrier Breaker 1.4 yang sudah kita siapkan sebelumnya. Setelah semua selesai maka pilihlah upgrade. Prosesnya akan berjalan 5-10 menit.



Gambar 4.2 Proses Upgrade firmware OpenWRT

- d) Setelah selesai proses upgrade, lakukan konfigurasi IP pada AP/router menjadi 192.168.34.1 dan simpan konfigurasi tersebut, serta lakukan reboot
- e) Jika tidak ada masalah saat proses upgrade, maka OpenWRT bisa diakses melalui browser melalui dengan IP 192.168.34.1. tampak pada gambar 4.3 .



Gambar 4.3 Tampilan awal firmware

4.1.2 Konfigurasi Open Voice Sistem

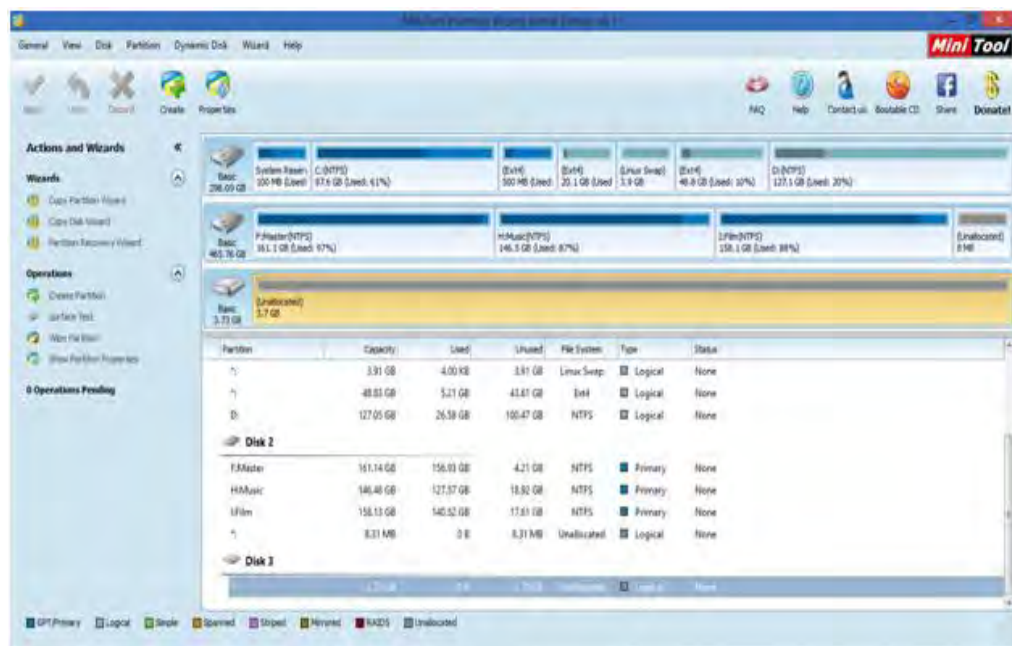
Untuk membuat sistem komunikasi yang berbasis Open-Voice dengan mengimplementasikan wireless VoIP, dan MANET maka diperlukan konfigurasi modul-modul yang sudah tertanam pada embedded device (router). Konfigurasi protokol pada sistem ini menggunakan OLSR sebagai routing protokolnya yang akan menghubungkan node-node yang ada dalam jaringan MANet. Serta konfigurasi codec yang di pilih jenis codec GSM

4.1.3 Install dan Konfigurasi Extroot

Pada proses konfigurasi ini diperlukan sebuah media storage flashdisk dengan kapasitas 16 GB, dengan kapasitas tersebut menjadikan lebih leluasa yang digunakan untuk menambah storage dan menyimpan modul-modul pada proses instalasi Asteris. Berikut tahapan proses dari extroot :

1. Proses Partisi Flashdisk

Sebelum dilakukan tahapan untuk extroot, terlebih dulu dilakukan partisi terhadap flashdisk menjadi dua bagian. Yaitu :
partisi pertama untuk *swap* dengan kapasitas sebesar 15GB dan partisi kedua sebagai *root* dengan kapasitas 1GB. Berikut gambar 4.4, menunjukkan proses partisi.



Gambar 4.4 Partisi Flashdisk

2. Proses login dan proses konfigurasi Extroot

Untuk memulai tahapan extroot, dilakukan login pada root. Gambar 4.5a. menunjukkan proses login pada level root. Langkah selanjutnya update dengan memberikan beberapa perintah pada proses extroot, ditunjukkan pada gambar 4.5b, dibawah ini.



(a)

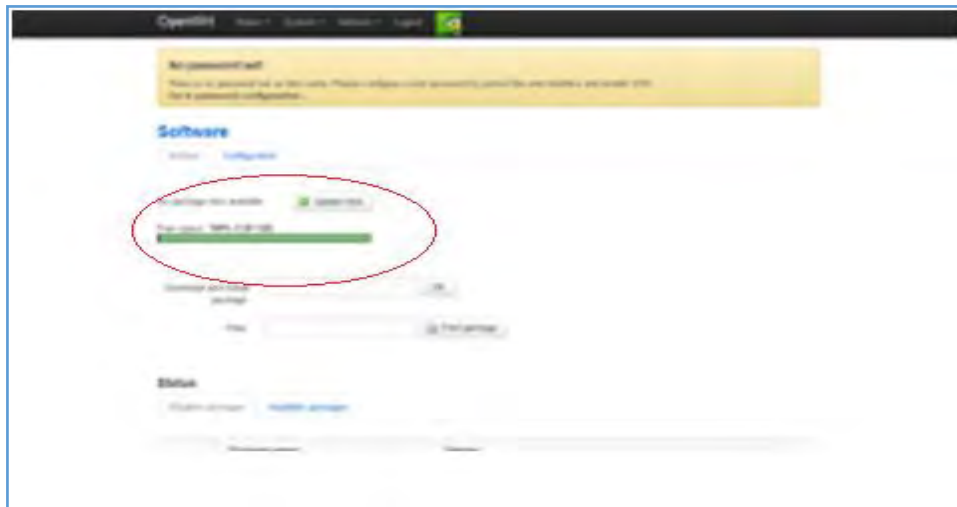


(b)

Gambar 4.5 (a) Login root dan (b) proses pemberian perintah extroot

3. Proses akhir extroot

Berikut adalah proses akhir extroot yang ditunjukkan oleh gambar 4.6 hasil extroot yang proses instalasi akan selesai (dengan tanda lingkaran warna merah)



Gambar 4.6 Proses akhir tahapan extroot

4.1.4 Instalasi Server VoIP

Pada tahapan ini dilakukan Instalasi asterisk pada sistem OpenWRT yang telah terembedded pada perangkat router pada tahapan sebelumnya diatas, sehingga perangkat router menjadi sebuah Server VoIP. Berikut adalah tahapan dalam instalasi asterisk yang perlu dilakukan.

- a) Langkah pertama adalah setup open instalasi asterisk beserta library codecnya yang akan dipakai dalam sistem. Instalasi dilakukan melalui service webbase yang dimiliki router ini agar lebih user friendly melalui repository offline yang telah dibuat sebelumnya untuk mempercepat proses instalasi paket.
- b) Setelah Instalasi selesai, langkah selanjutnya adalah konfigurasi asterisk yang berada di /etc/asterisk menggunakan aplikasi *winscp*
- c) Tahapan selanjutnya akan dilakukan konfigurasi OLSR routing protocol yang berfungsi sebagai protocol utama dalam mengatur aliran data. Karena dalam firmware ini sudah diikutkan dengan preconfigure file dari olsrd pada saat proses kompilasi firmware maka yang perlu dilakukan hanya menyesuaikan dengan ip address yang digunakan oleh interface yang digunakan.

4.1.5 Konfigurasi Native Bridging

Tahapan konfigurasi ini untuk mengubah mode awal Generic Bridging menjadi Native Bridging, dengan melakukan konfigurasi setting value pada dua file yaitu *sip.conf* dan *extension.conf*.

File Sip.conf memuat variabel Canreivite dengan nilai “Yes/No”, ini akan berfungsi untuk memotong atau lalu lintas RTP paket suara tidak melalui inti asterisk

File Extension.conf adalah file yang berisi aturan routing panggilan yang terjadi, jika asterisk akan melayani pengguna. Konfigurasi lengkap yang digunakan dalam penelitian ini terlampir.

Berikut dibawah ini adalah potongan konfigurasi native bridging pada file sip.conf dan Extension.conf.

a. **File Sip.Conf :**

```
[001]                : nomor call Client-1
type=friend
context=my-phones
username=001
secret=001
host=dynamic
nat=no               ;
canreivite= yes      ; memungkinkan RTP lalu lintas suara untuk
                     ; memotong Asterisk

disallow=all
allow=gsm             ; setting codes GSM
dtmfmode=info
```

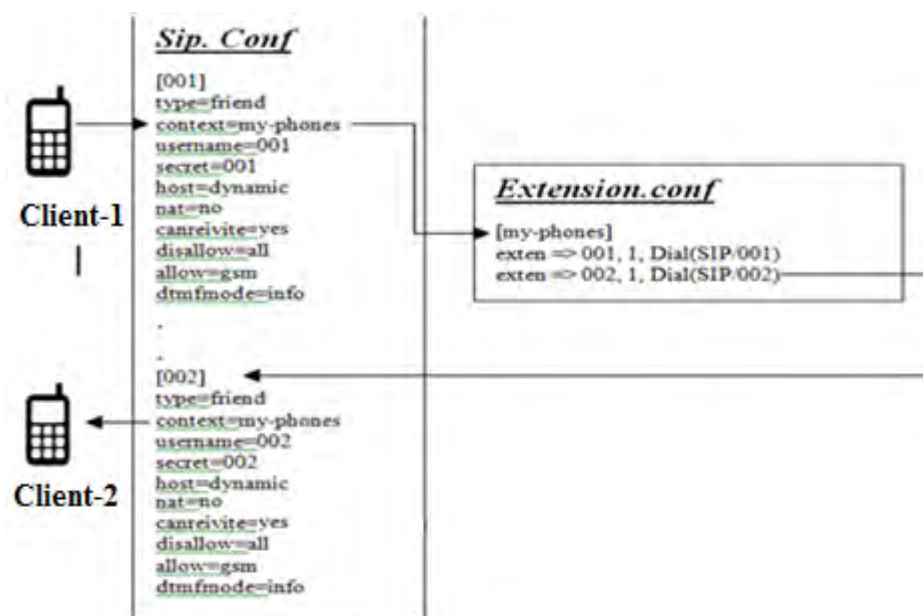
```
[002]                ; nomor call Client-2
type=friend
context=my-phones
username=002
secret=002
host=dynamic
nat=no               ;
canreivite= yes      ; memungkinkan RTP lalu lintas suara untuk
                     ; memotong Asterisk          ;

disallow=all
allow=gsm             ; setting codes GSM;
dtmfmode=info
```

b. **File extesion.conf:**

```
[my-phones]
exten => 001, 1, Dial(SIP/001)
exten => 002, 1, Dial(SIP/002)
```

.konfigurasi tersebut diatas, memberikan nilai *Canreivite= yes* ; dan *allow=gsm* , dengan konfigurasi tersebut diatas digunakan ketika kondisi reINVITEs yang diaktifkan. Dengan konfigurasi native bridging ini, data audio mengalir di luar inti server (Asterisk) pada Server VoIP. sehingga endpoint mengirimkan media suara mereka langsung satu sama lain antar endpoint. Dengan syarat antara client telah diseting dengan codec yang sama yaitu GSM. Gambar 4.7, memperlihatkan urutan proses pemanggilan antar client-1 ke client-2 dengan konfigurasi native bridging.



Gambar 4.7 Urutan proses pemanggilan

4.2 Data Hasil Pengujian

Data hasil pengujian dan pengukuran baik itu berupa angka pada tabel dan data capture hasil pengujian, yang telah dilakukan sesuai dengan bahasan pada bab sebelumnya. Pengambilan datanya dengan menggunakan software analyzer Wireshark version 1.12.4

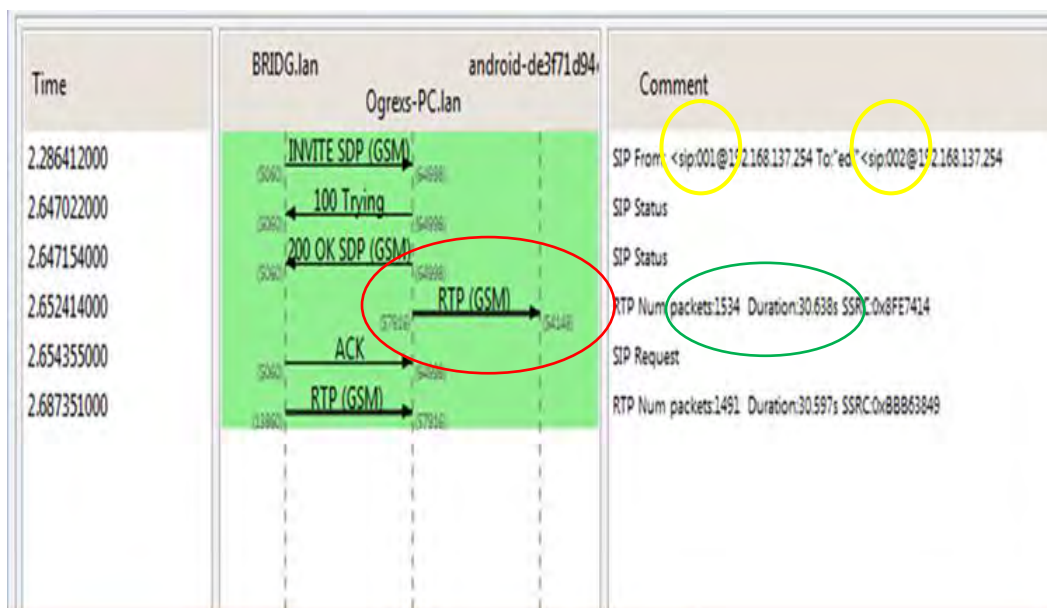
4.2.1 Data Pengujian Panggilan pada Sistem

Pada pengujian dan pengukuran panggilan pada sistem, dilakukan dua pengujian, yaitu pengujian pada sisten server voip dan pengujian server voip atau node dengan penentuan jarak panggilan

a) Pengujian Sistem Server VoIP

Langkah awal pengujian sistem server voip, dengan cara mengkoneksikan antar client-1 dan client-2 ke node server voip, melalui wifi pada smartphone masing-masing client ke wifi node server. Untuk memastikan kedua client terkoneksi ke server/node, dapat dilihat pada server melalui fasilitas list network pada editor sisten OpenWRT. Akan terlihat beberapa client sudah terkoneksi dengan Server dan diantaranya sudah mendapatkan IP dari Node/Server VoIP.

Langkah berikutnya dilakukan pengujian dengan melakukan panggilan dari client-1 ke client-2 dengan menggunakan aplikasi Zoiper. Gambar 4.8 menunjukkan panggilan dapat diterima dan terjadi percakapan antara client-1(sip001) sebagai pemanggil dengan client-2 (sip002) sebagai penerima (dengan tanda lingkaran warna kuning). Terjadinya percakapan (dengan tanda lingkaran warna merah) dengan durasi 30,638 detik serta 1534 paket terkirim (dengan tanda lingkaran warna hijau), sebagai petunjuk konfigurasi native bridging berjalan dengan baik.



Gambar 4.8 Pengujian call client-001 ke client-002

b) Pengujian dengan jarak panggilan

Dari pengukuran jarak antara node dengan client didapatkan data seperti yang tertera pada Tabel 4.1. dimana telah dilakukan pengujian sekaligus pengukuran pada jarak awal 10 meter, dengan kelipantanya hingga jarak 100 meter lebih. Sehingga diperoleh selain jarak sinyal yang dapat dijangkau untuk menentukan area cakupan, juga diukur kualitas nilai throughput, delay, dan paket loss serta waktu.

Tabel 4.1 Hasil pengukuran jarak antara node dengan client

Jarak (mtr)	Throughput(bps)	Delay(ms)	Paket loss (%)	Waktu (menit)
10	0,032	49,334	0	00,49
20	0,003	48,172	0	00,48
30	0,024	64,399	0	01,09
40	0,006	30,566	0	00,41
50	0,004	37,741	0	00,45
60	0,029	46,207	0	00,46
70	0,029	46,197	0	00,46
80	0,041	36,473	0	00,40
90	0,019	46,504	0	00,46
100	0,018	177,696	0	02,57
100+	0,004	66,096	-400%	01,06

4.2.2 Data Pengujian Kapasitas Maksimum Panggilan Per Node

a. Pengujian dan pengukuran kapasitas panggilan dengan genric bridging

Berikut adalah data hasil pengukuran kapasitas yang bisa dilayani per node dengan menggunakan konfigurasi generic bridging. Tabel 4.2 terlihat jumlah maksimal pasangan client yang mampu di layani oleh server voip, berjumlah 11 pasang atau 22 client.

Tabel 4.2 Data kapasitas layanan per node dengan generic bridging

Urutan Client	Throughput (Mbps)	Delay (ms)	Packet Loss %	Codec	Durasi Panggilan (m)
1	0,034	344,035	0	GSM FR	01:19
2	0,034	358,535	0	GSM FR	03:18
3	0,033	382,277	0	GSM FR	05:06
4	0,033	383,178	0	GSM FR	06:57
5	0,032	384,394	0	GSM FR	07:15
6	0,032	393,190	0	GSM FR	08:29
7	0,032	393,976	0	GSM FR	09:19
8	0,032	353,813	0	GSM FR	10:45
9	0,032	737,119	0	GSM FR	12:25
10	0,031	1,028,352	0,7	GSM FR	13:07
11	0,030	1,122,670	1.3	GSM FR	15:12

b. Pengujian dan pengukuran kapasitas panggilan dengan native bridging.

Pada tabel 4.3 merupakan data hasil pengujian dan pengukuran dengan konfigurasi native bridging, jumlah client yang berhasil melakukan pemanggilan dan percakapan adalah sebanyak 13 pasang, ini menunjukkan jumlah pasangan maksimal panggilan yang dapat di dilayani oleh server voip. Dan apabila dilakukan pemanggilan oleh pasangan client berikutnya ke-14 panggilan akan dilayani, tetapi dari salah satu pasangan client pemanggilan sebelumnya terputus. Ini menunjukkan kemampuan server voip maksimal melayani panggilan sebanyak 13 pasang client (26 user).

Tabel 4.3 Hasil pegujian kapasitas layanan per node dengan native bridging

Urutan Client	Throughput (Mbps)	Delay (ms)	Pkt Loss (%s)	Codec	Waktu (menit)
1	0,062	86,008	0	GSM FR	01:26
2	0,051	324,124	0	GSM FR	03:24
3	0,049	353,694	0	GSM FR	05:54

4	0,049	395,243	0	GSM FR	06:36
5	0,049	455,693	0	GSM FR	07:47
6	0,048	497,623	0	GSM FR	08:16
7	0,048	567,793	0	GSM FR	09:28
8	0,047	637,953	0	GSM FR	10:38
9	0,046	739,383	0	GSM FR	12:20
10	0,046	764,155	0	GSM FR	13:45
11	0,045	853,341	0	GSM FR	15:24
12	0,046	802,232	0	GSM FR	16:44
13	0,033	879,109	0	GSM FR	17:35

4.2.3 Data Pengujian Koneksi dan Kapasitas antar Node

Pada pengujian koneksi antar node dilakukan dua pengukuran. Pengujian pertama adalah koneksi antar node dan pengujian kedua mengukur kapasitas panggilan antar node, berikut adalah tabel data :

a) Pengujian kualitas koneksi antar node

Pengujian koneksi dilakukan pemanggilan dan dilanjutkan dengan percakapan antar client yang terkoneksi di masing-masing node, dilakukan sebanyak 3 kali pengujian, dengan hasil pengukuran tertera pada tabel 4.4, dari tabel tersebut ditunjukkan nilai throughput dari server maupun client, nilai paket loss serta jitter.

Tabel 4.4 Hasil pengukuran kualitas koneksi antar node

Pengujian	Throughput (bps)		Paket Loss (%)	Jitter (ms)	Waktu (menit)
	Server	Client			
1	209,826	329.00	0	20	02:47
2	290,368	177.00	0	20	02:54
3	258,420	145.00	0	21	02:16

b) Pengujian kapasitas panggilan antar node

Pengujian kapasitas panggilan antar node dengan menambah node-2 pada sisten jaringan. Hasil pengukuran kapasitas yang dapat dilayani oleh server sejumlah 10 pasang atau 20 client. Tabel 4.5 memaparkan hasil pengukuran jumlah client, nilai throughput, nilai delay dan nilai paket loss.

Tabel 4.5 Data pengukuran kapasitas panggilan antar node

Urutan Client	Throughput (Mbps)	Delay (ms)	Paket Loss (%)	Payload /Codec	Waktu (menit)
1	0,047	354,36	0	GSM FR	01:39
2	0,048	330,99	0	GSM FR	02:27
3	0,047	434,37	0	GSM FR	03:21
4	0,046	507,45	0	GSM FR	04:28
5	0,045	700,59	0	GSM FR	05:41
6	0,045	824,16	0	GSM FR	07:35
7	0,046	872,62	0	GSM FR	08:13
8	0,043	964,59	0	GSM FR	09:42
9	0,032	1.021,56	0	GSM FR	10:39
10	0,025	1.450,26	0	GSM FR	12:09

4.3 Analisa Data Pengujian

Data hasil pengukuran pada pengujian ini akan ditampilkan dalam bentuk format grafik, untuk memudahkan analisa kualitas VoIP.

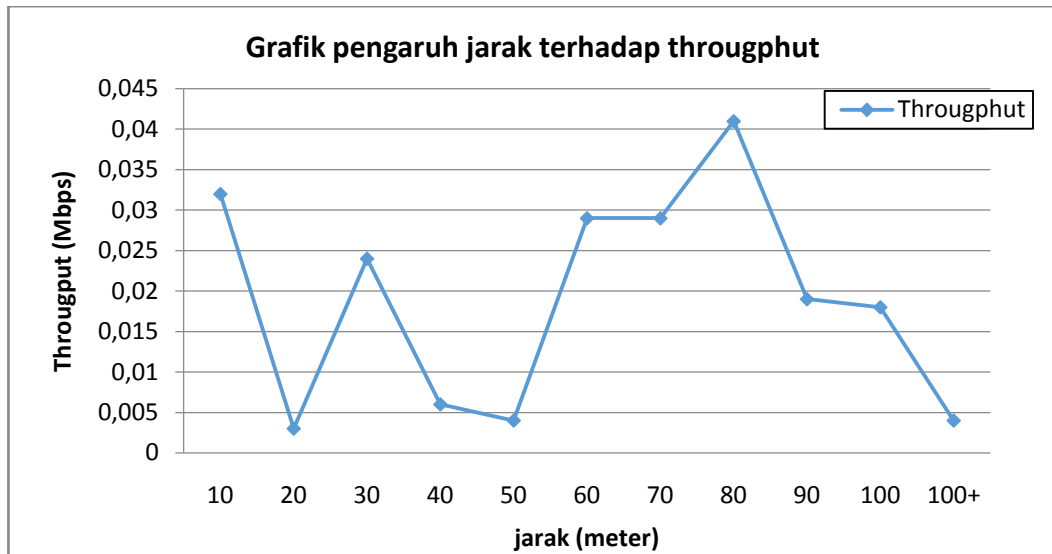
4.3.1 Analisa Data Pengujian Call pada Sistem

Untuk analisa data hasil pengukuran dan pengujian pemanggilan terhadap sistem terdapat 3 variabel pengukuran, yaitu :

a) *Throughput*

Grafik gambar 4.9 menunjukkan nilai throughput yang tidak stabil antara jarak pengukuran 10-60 meter. Pada jarak 80 meter nilai throughput pada titik tertinggi.

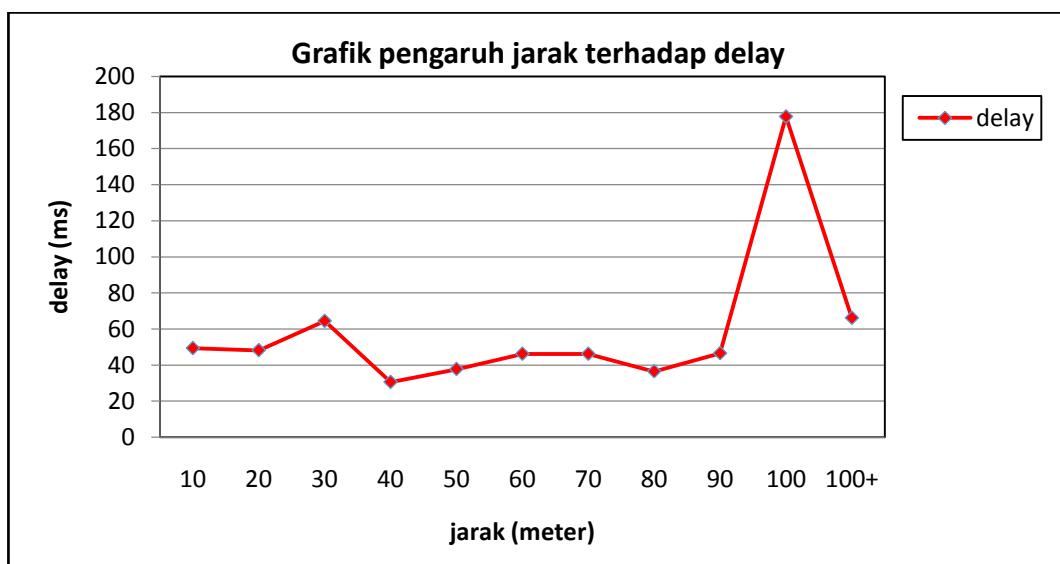
Pada jarak 90-100 meter lebih nilai throughput melemah, disebabkan daya tangkap sinyal node semakin lemah karena jarak semakin jauh.



Gambar 4.9 Grafik pengaruh jarak terhadap troughput

b) Delay

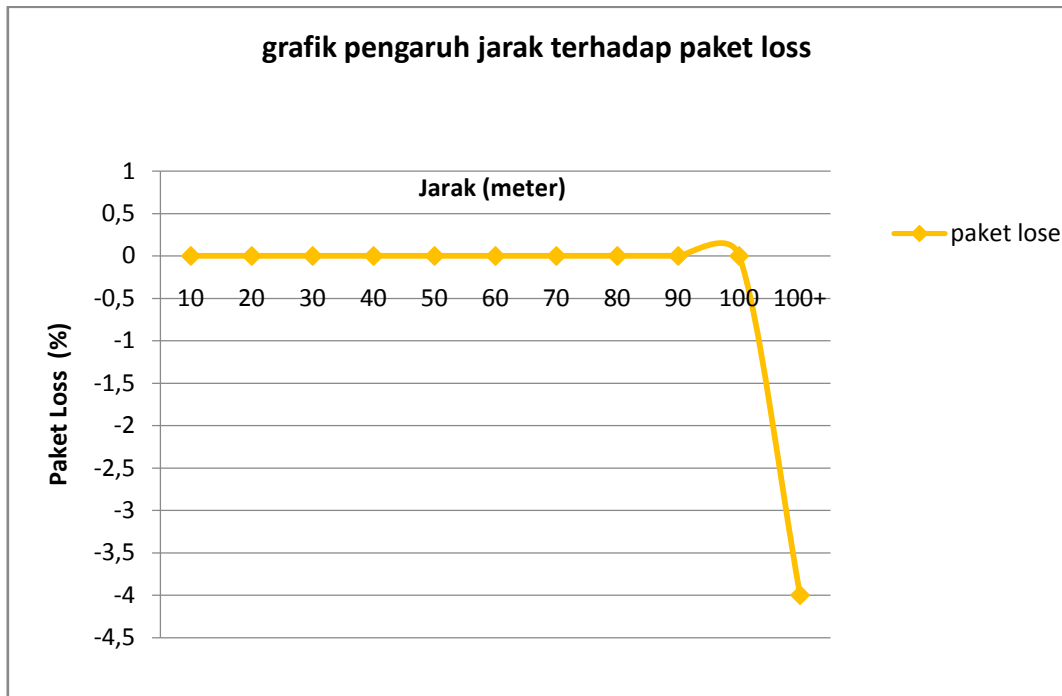
Terlihat pada gambar 4.10 bahwa secara umum nilai delay yang diterima memiliki trend yang cenderung meningkat, berbanding lurus dengan pertambahan jarak yang terjadi. Nilai delay tertinggi pada posisi jarak 100 meter. Hal ini karena semakin lamanya waktu yang diperlukan untuk melakukan pengiriman data yang menyebabkan delay yang semakin tinggi pula..



Gambar 4.10 Grafik pengaruh jarak terhadap delay

c) Paket Loss

Grafik yang ditampilkan pada gambar 4.11, dapat dilihat bahwa nilai paket loss cenderung stabil pada jarak pengukuran 10 meter sampai 90 meter, dan pada jarak melebihi 100 meter, paket loss berkurang secara signifikan hal ini terjadi karena dipengaruhi nilai delay turun



Gambar 4.11 Grafik pengaruh jarak terhadap paket loss

4.3.2 Analisa Data Kapasitas Maksimum Panggilan Per Node

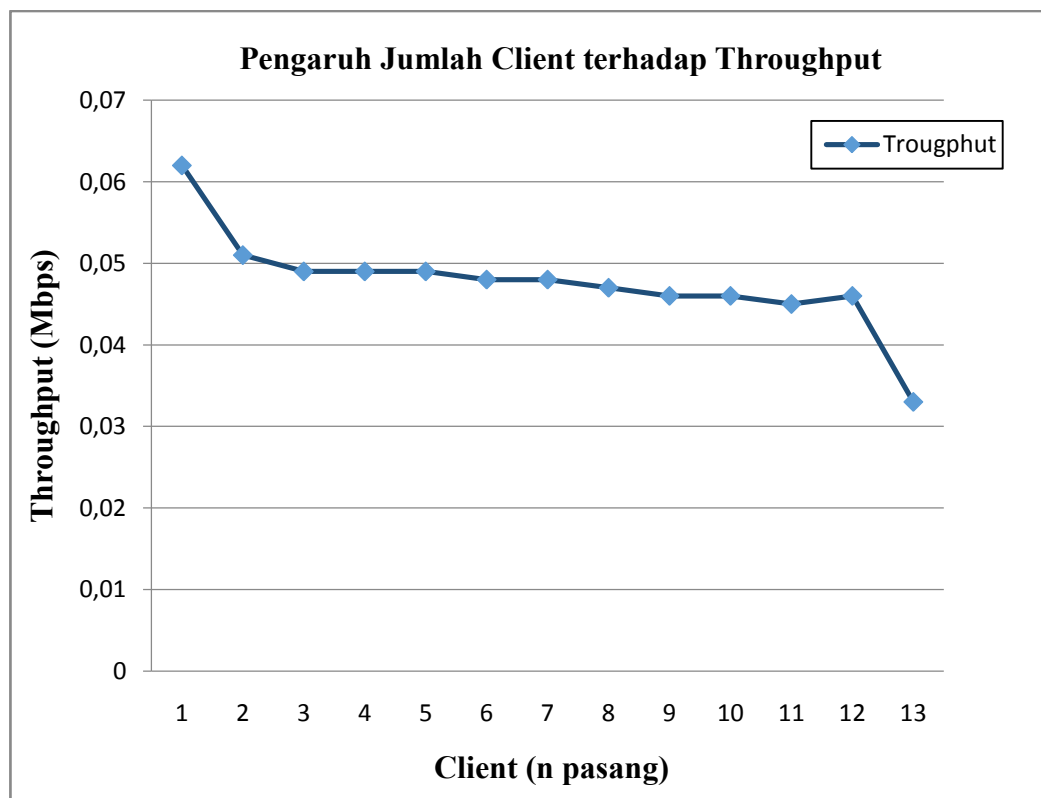
Pada analisa data kapasitas panggilan per node terbagi menjadi dua, yaitu analisa kualitas data hasil pengukuran kapasitas panggilan dengan konfigurasi native bridging dan analisa perbandingan antara data kapasitas panggilan dengan metode native dan generig bridging.

Berikut adalah analisa meliputi kualitas nilai throughput dan delay dari data hasil pengukuran kapasitas panggilan per node dengan konfigurasi native bridging.

a) Throughput

Pada grafik gambar 4.12, menunjukkan pasangan terakhir yang melakukan aktifitas pemanggilan adalah pasangan client-13. Saat pasangan selanjutnya

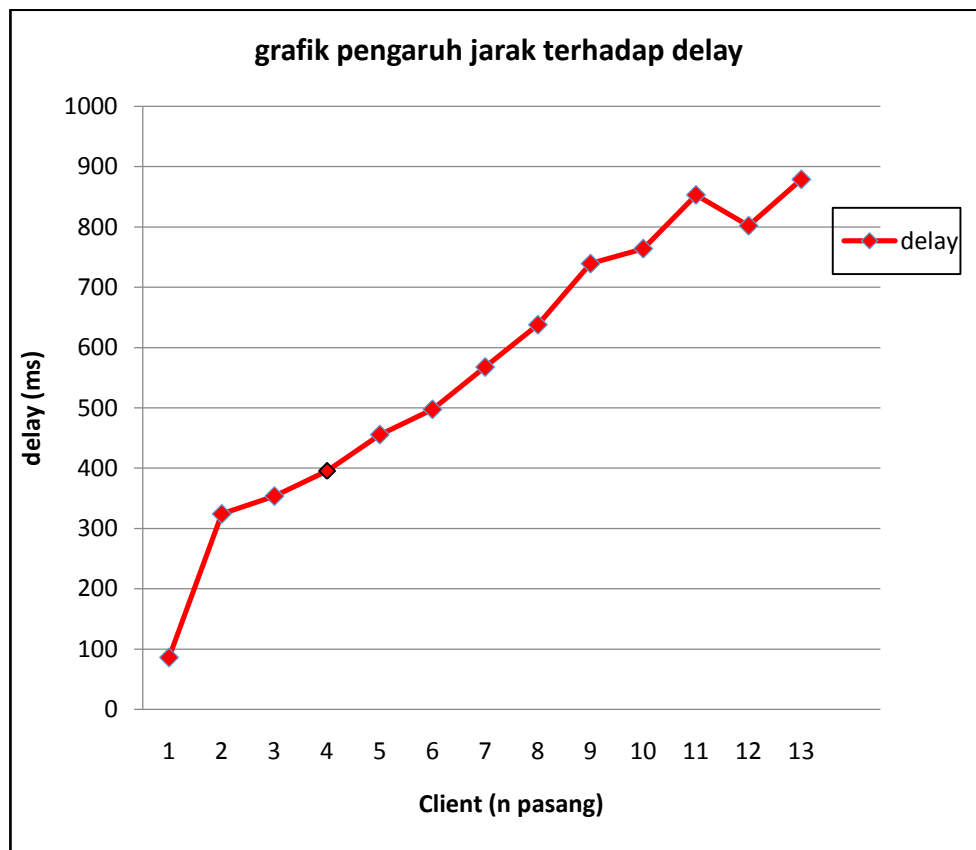
client-14 melakukan panggilan masuk ke dalam sistem mengakibatkan salah satu pengguna sebelumnya terputus dari sistem. Hal ini dikarenakan dengan bertambahnya pengguna yang terkoneksi ke dalam sistem maka semakin berkurang juga throughput yang di dapat. sehingga berdampak pada besar data yang bisa dikirim berkurang. Dan dapat di analisa dalam pengujian didapatkan hasil bahwa dengan bertambahnya client, menjadikan throughput yang diperoleh cenderung semakin kecil dikarenakan berbagi saluran.



Gambar 4.12 Pengaruh jumlah client terhadap throughput

b) Delay

Grafik yang tersaji pada gambar 4.13, dapat dilihat bahwa nilai delay mengalami kecenderungan meningkat, ini berbanding lurus dengan penambahan jumlah client yang melakukan pemanggilan. Hal ini dikarenakan dengan bertambahnya jumlah aktifitas pemanggilan, maka delay cenderung akan semakin besar.



Gambar 4.13 Pengaruh jumlah client terhadap delay

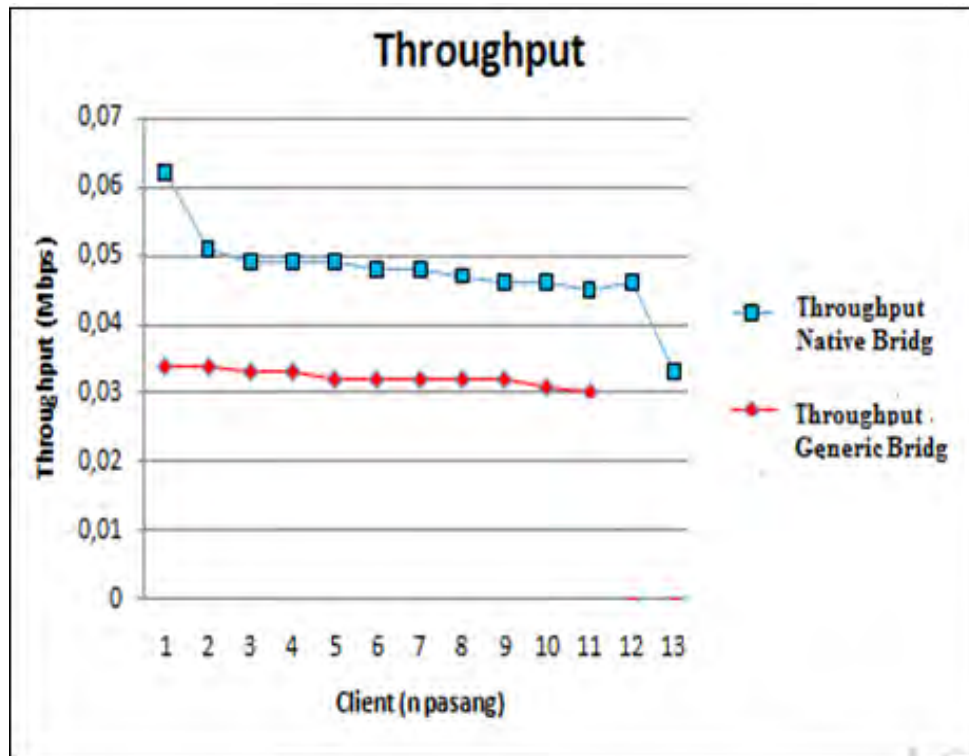
Berikut adalah analisa perbandingan antara data kapasitas panggilan dengan metode native bridging dan generic bridging. Analisa perbandingan ini selain membandingkan jumlah kapasitas panggilan, juga menganalisa kualitas nilai throughput dan delay pada kedua konfigurasi.

Jumlah maksimal kapasitas panggilan pada native bridging berjumlah 26 client (13 pasang), sedangkan jumlah maksimal kapasitas panggilan pada generic bridging berjumlah 22 client (11 pasang). Terjadi peningkatan sebesar 18,18% , berikut untuk analisa pada kualitas troughput dan delay dari kedua konfigurasi.

a) Throughput

Pada gambar 4.14. grafik terlihat dengan jumlah panggilan yang sama, nilai throughput untuk metode native bridging lebih tinggi, dengan nilai diatas 0,04 Mbps dibandingkan dengan nilai throughput untuk generic bridging

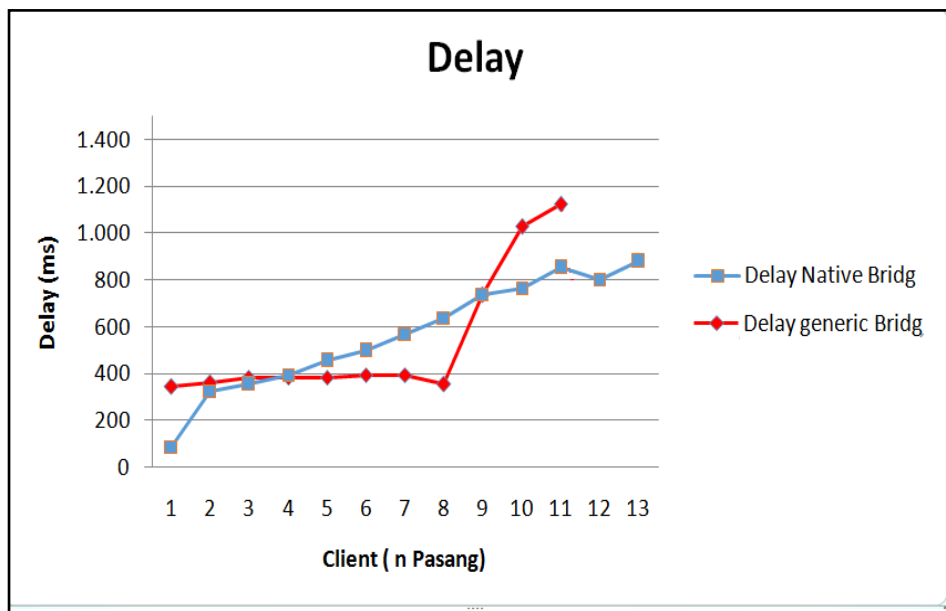
yang rata-rata dibawah 0.040 Mbps. Ini menunjukan dengan menggunakan metode native bridging terjadi peningkatan nilai throughput pada sistem komunikasi voip yang digunakan.



Gambar 4.14 Perbandingan Throughput pada native dan generic bridging

b) Delay

Grafik nilai delay pada gmbar 4.15, menunjukkan nilai delay baik itu untuk native maupun generic bridging sama-sama mengalami peningkatan. Tetapi delay pada native briging diawali dengan nilai delay yang kecil yaitu 86,08 ms dan siring dengan jumlah client yang masuk, delay mengalami kenaikan tapi stabil. Sedangkan delay pada generic bridging diawali dengan nilai diatas 300 ms dan terjadi peningkatan yang signifikan hingga 1000 ms pada saat masuknya client ke 8 sampai terakhir. Jadi pada pada saat urutan sama-sama client ke 8, delay pada konfigurasi native bridging lebih rendah dibanding dengan delay pada konfigurasi generic bridging.



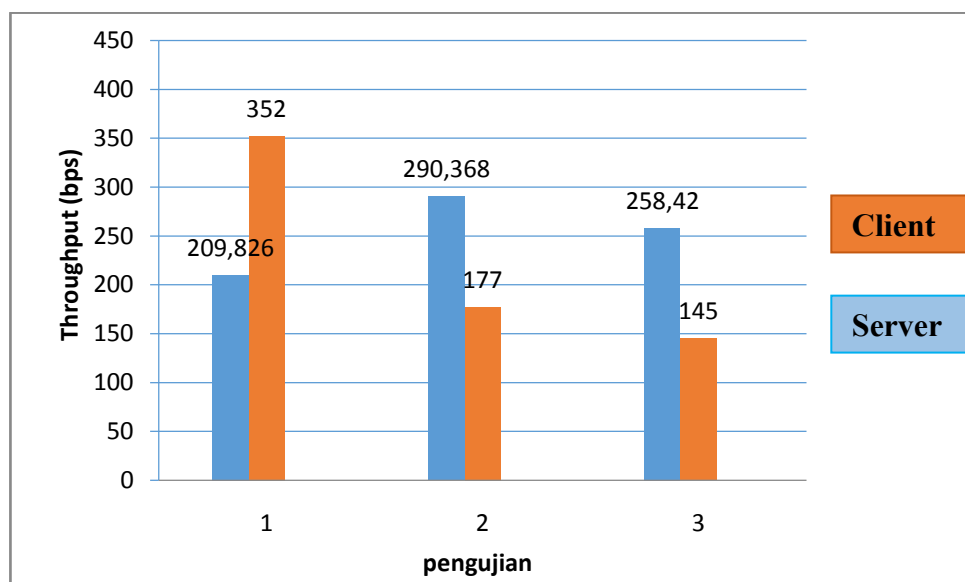
Gambar 4.15 Perbandingan Delay pada native dan generic bridging

4.3.3 Analisa Data Pengujian Koneksi dan Kapasitas Panggilan Antar Node

Pada analisa data koneksi antar node terbagi menjadi dua yaitu analisa data hasil pengukuran koneksi antar node dan analisa kapasitas panggilan antar node.

Berikut adalah analisa meliputi kualitas throughput dan delay dari data hasil pengukuran koneksi antar node, yang dilakukan sebanyak 3 kali pengukuran, dilakukan untuk :

a) Throughput

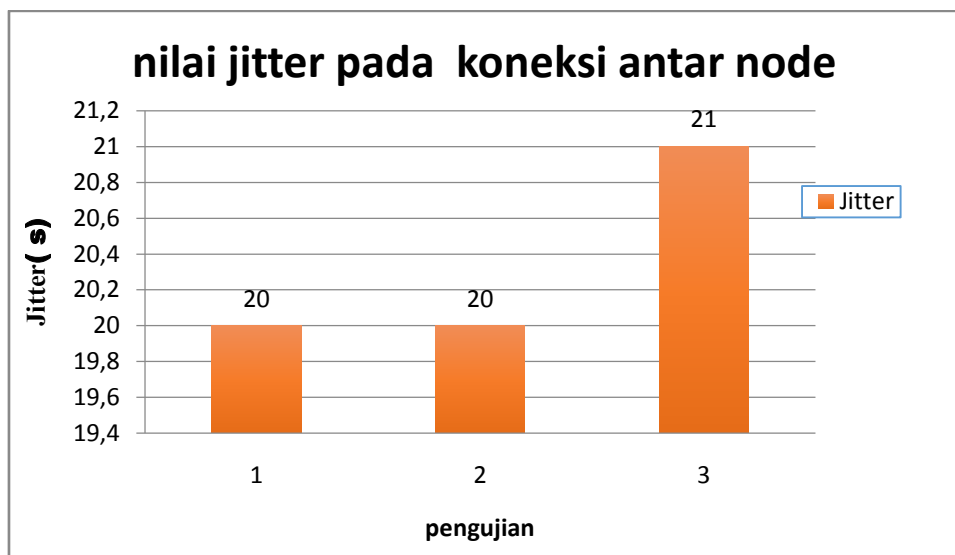


Gambar 4.16 . Throughput pada pengujian koneksi antar Node

Pada Gambar 4.16. dapat dilihat, throughput pada server pada tiga kali pengujian mempunyai nilai rata-rata diatas 200bps, kenaikan maupun penurunan throuhput tidak signifikan, masih stabil. Untuk throughput pada client pada pengujian pertama mempunyai nilai yang tinggi yaitu 352bps, tapi pada pengujian ke dua dan tiga mengalami penurunan hingga setengahnya. Ini dikarenakan dari pengujian dua dan tiga nilai througput pada server mengalami kenaikan. Jika di amati terdapat karakreistik pada tiap pengujian nilai throughput server lebih tinggi atau rendah dari client dan sebaliknya. Ini menunjukkan kualitas koneksi pada implementasi sistem jaringan masih dalam kategori baik,

b) Jitter

Dari ketiga pengujian dan pengukuran, terlihat pada gambar 4.17, menunjukkan untuk pengujian ke satu dan dua nilai jitter sebesar 20ms. Ini nenjuukan bahwa kualitas jitter pada kriteria baik. Sedangkan pada percobaan ketiga naik menjadi 21ms, kualitas jitter tergolong pada kreteria cukup. Dari ketiga pengujian nilai jitter masih dibawah ambang batas toleransi yang dapat diterima untuk komunikasi yaitu $> 50\text{ms}$ dengan kulitas buruk atau tidak dapat didengar. Sehingga dengan kualitas jitter antar baik dan cukup tersebut, bisa dikatakan kualitas komunikasi masih layak untuk diimplemetasikan pada sistem jaringan.

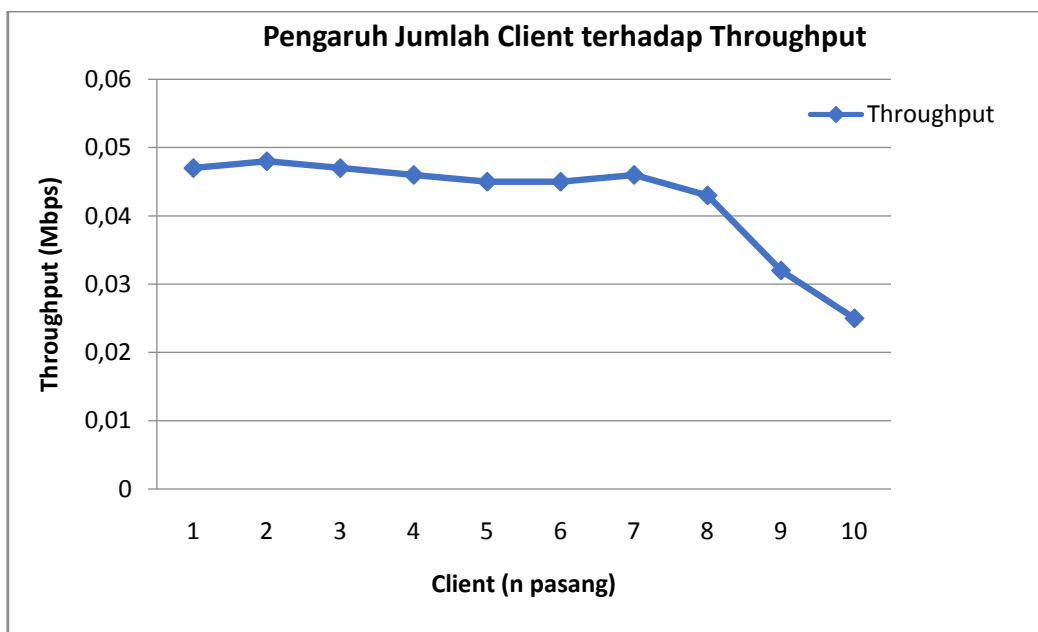


Gambar 4.17 Jitter pada pengujian koneksi antar node

Berikut ini adalah analisa data hasil dari pengukuran kapasitas maksimal panggilan antar node. Pada grafik gambar 4.16 dan 4.17, menunjukkan pasangan terakhir yang melakukan aktifitas pemanggilan adalah pasangan client-10. Saat pasangan selanjutnya client-11 melakukan panggilan masuk kedalam sistem mengakibatkan salah satu pengguna sebelumnya terputus dari system. Terputusnya salah satu client, dikakrenakn nilai throughput yg terus menurun dan delay semakin besar. Berikut analisa nilai kualitas throughput dan delay.

a. Throughput

Tampak pada gambar 4.18. Throughput semakin menurun, hal ini dikarenakan dengan bertambahnya pengguna yang terkoneksi kedalam sistem maka semakin berkurang juga throughput yang di diperoleh. Sehingga berdampak pada besar data yang bisa dikirimkan berkurang. Dan dapat di analisa dalam pengujian didapatkan hasil bahwa dengan bertambahnya client, menjadikan throughput yang diperoleh cenderung semakin kecil. Semakin kecilnya nilai throughput ini dikarenakan berbagi saluran.

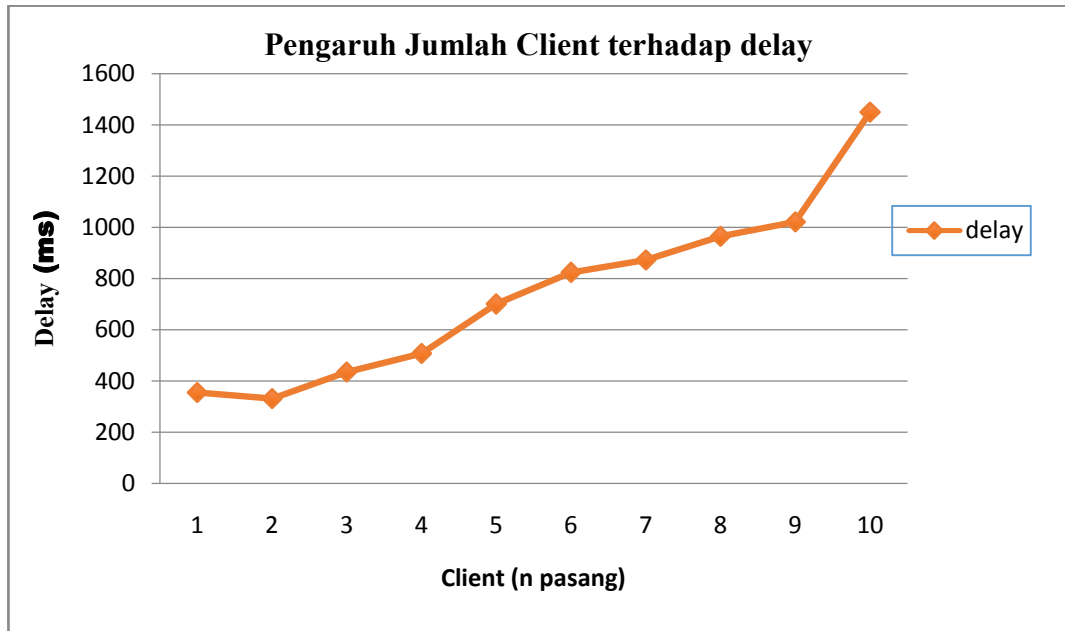


Gambar 4.18 Throughput pada kapasitas panggilan antar node

b. Delay

Grafik yang tersaji pada gambar 4.19, dapat dilihat bahwa nilai delay mengalami kecenderungan meningkat, ini berbanding lurus dengan penambahan

jumlah client yang melakukan pemanggilan. Peningkatan delay ini dikarenakan bertambahnya jumlah aktifitas pemanggilan, berakibat bertambah banyaknya paket yang dikirim ketujuan. maka banyaknya paket tersebut mempengaruhi delay cenderung akan semakin besar



Gambar 4.19 Delay pada kapasitas panggilan antar node

!-- Halaman ini sengaja di kosongkan --!

BAB 5

KESIMPULAN DAN SARAN

Dari hasil pengujian dan pengukuran serta analisa diatas dapat ditarik beberapa kesimpulan, beserta permasalahan yang bisa dibahas sebagai kelanjutan dari penelitian ini.

5.1 Kesimpulan Penelitian

1. Perancangan dan implementasi wireless VoIP pada embedded device sebagai server VoIP dengan menggunakan metode native bridging dapat berjalan dengan baik dengan komunikasi singel node maupun multi node, ditandai dengan koneksi client ke node. dan client dapat melakukan panggilan ke client lain serta adanya percakapan antar client.
2. Pada jarak jangkauan pemanggilan 10-90 meter, kualitas suara yang diterima oleh pengguna kriteria kualitas baik, karena nilai delay dibawah batas maksimum 150 ms, sedangkan jitter dan paket loss juga mempunyai kriteria kualitas baik. Kualitas tersebut berdasarkan rekomendasi dari ITU-T G.114
3. Sedangkan pada jarak jangkauan melebihi 100 meter, dengan throughput 0,018 Mbps, delay 177,696 ms, paket loss 0% dan jitter 0,0 ms. Dapat disimpulkan secara keseluruhan kualitas voice yang di terima, pada kategori cukup, karena nilai delay masih pada dibawah batas maksimal yaitu 250ms. Sesuai dengan yang direkomendasikan oleh ITU-T G.114
4. Konfigurasi native bridging dapat berjalan dengan baik, dengan dapat diterimanya panggilan dan terjadi proses percakapan antar client sejumlah 13 pasang atau 26 client. Kapasitas panggilan terjadi peningkatan, dibanding dengan server voip yang menggunakan konfigurasi generic bridging yang hanya mempunyai kapasitas panggilan

11 pasang atau 22 client. Peningkatan kapasitas panggilan tersebut sebesar 18,18%.

5. Dengan menambah node untuk menambah jarak jangkauan, kualitas koneksi antara server dan client masih dikatakan layak. Dengan nilai throughput pada server rata sebesar 252,87bps dan pada client rata-rata 224,66bps. Kualitas jitter diantara baik dan cukup, dibawah batas toleransi.
6. Sedangkan kapasitas panggilan yang mampu dilayani oleh server berjumlah 10 pasang atau 20 client. Kapasitas panggilan mengalami penurunan sebesar 23,08%, dibandingkan dengan kapasitas panggilan per node dengan metode native bridging yang sama. Penurunan kapasitas tersebut disebabkan panjang atau jauhnya rute jarak panggilan antar client yang harus melewati dua node.

5.2 Saran

Dari perancangan dan pengujian yang dilakukan masih jauh dari sempurna. Dan masih banyak yang perlu dikembangkan untuk memaksimalkan kinerja dari sistem, diantaranya :

1. Melakukan riset kelanjutan terhadap perangkat AP/router type lain dengan menggunakan metode native bridging
2. Melakukan penelitian server voip dengan metode native bridging yang aplikasikan dengan beberapa protokol routing dan jenis codec yang berbeda.

DAFTAR PUSTAKA

- Asriadi , Istas Pratomo,(2014) ” WiFi BTS : Konsep dan Desain Komunikasi Nirkabel Bergerak Untuk Rural Area Berbasis OpenWRT “, Prosiding Seminar SISTI 2014 ISBN 978 – 979 – 3288 – 92. Institut Teknologi Sepuluh November .
- Dimas Lazuardi,(2008) ”Analisa Kinerja Implementasi Wireless Distribution system pada Perangkat Access 802.11 G Menggunakan OpenWRT", Institute Teknologi Sepuluh November Surabaya,
- Effan Najwaini,(2014) “Analisis Kinerja Voip Server pada Wireless Access Point “, Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta
- G.114, ITU-T. Retrieved Desember 10, 2014, from <http://www.itu.int/rec/T-REC-G.114/en>
- Ganguly, S., Navda, V., Kim, K., & Kashyap, A. (2006). “Performance optimizations for deploying voip services. *Selected Areas in Communications*”, *IEEE Journal on* , 2147 - 2158 .
- Hrituparna PaulA, Priyanka Sarkar,(2013) “ Studi and Comparison OLSR, AODV and ZRP Routing Protokcols in Ad Hoc Networks “, *National Institute of Technology, Agartala*,
- Istas P., Asriadi, Ach. Affandi,(2013) “Menerapkan Sederhana dan Rendah-Biaya Ponsel VoIP Server Berbasis Pada Open Source Wireless Router “Dep. Teknik Elektro, Fakultas Teknologi Industri ITS Surabaya
- OpenWrt, "OpenWrt Wireless Freedom," (2012). [Online]. Available: <http://wiki.openwrt.org/doc/howto/voip.overview?s=kamailio>. [Accessed 9 Maret 2014].
- OpenWRT. (n.d.). Retrieved September 1, 2014, from <http://www.openwrt.org>

- Risky Agri Syafindra,(2010) “Analisa Performansi dan Kualitas Kanal VoIP Pada Sistem Embedded Wireless Berbasis 802.11 G”, Institut Teknologi Sepuluh Nopember,
- Samrat Ganguly, Vishnu Navda, (2006) ,“ Performance Optimizations for Deploying VoIP Services in Mesh Networks, IEEE Journal on Selected Area in Communications, 24, NO. 11,
- Seputra, Wahyu Edy, (September 2013) “ Perbandingan Kinerja Protokol AODV dengan OLSR pada MANET”, Semarang: Universitas Diponegoro
- Anonymous. (n.d.). *Asterisk*. Retrieved Desember 12, 2014, from <http://www.asterisk.org>
- Seto, K., & Ogunfunmi, T. (2013). Scalable Speech Coding for IP Networks: Beyond iLBC. *Audio, Speech, and Language Processing, IEEE Transactions on* , 2337 - 2345.

HALAMAN LAMPIRAN

7.1 Konfigurasi Penambahan Kapasitas memori

----- Extroot Configuration -----

```
#opkg install block-mount kmod-fs-ext4 kmod-usb-storage kmod-usb-ohci
#reboot
#block info
#mount /dev/sda1 /mnt
#mkdir /tmp/cproot
#mount --bind / /tmp/cproot
#tar -C /tmp/cproot -cvf - . | tar -C /mnt -xvf -
#sync ; umount /mnt
#umount /tmp/cproot
#block detect > /etc/config/fstab
#vi /etc/config/fstab
#/ect/init.d/fstab enable
#/etc/init.d/fstab start
#reboot
#df -h
```

7.2. Konfigurasi Native Bridging

File Sip. Conf

.....

[001]

type=friend

context=my-phones

username=001

secret=001

host=dynamic

nat=no

canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk

disallow=all

allow=gsm

dtmfmode=info

[002]

type=friend

context=my-phones

username=002

secret=002

host=dynamic

nat=no

canreivite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[003]
type=friend
context=my-phones
username=003
secret=003
host=dynamic
nat=no
canreivite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[004]
type=friend
context=my-phones
username=004
secret=004
host=dynamic
nat=no
canreivite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[005]
type=friend
context=my-phones
username=005
secret=005
host=dynamic
nat=no
canreivite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[006]
type=friend
context=my-phones

username=006
secret=006
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[007]
type=friend
context=my-phones
username=007
secret=007
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[008]
type=friend
context=my-phones
username=008
secret=008
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[009]
type=friend
context=my-phones
username=009
secret=009
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[010]
type=friend
context=my-phones
username=010
secret=010
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[011]
type=friend
context=my-phones
username=011
secret=011
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[012]
type=friend
context=my-phones
username=012
secret=012
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[013]
type=friend
context=my-phones
username=013
secret=013
host=dynamic
nat=no

canreivite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[014]
type=friend
context=my-phones
username=014
secret=014
host=dynamic
nat=no
canreivite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[015]
type=friend
context=my-phones
username=015
secret=015
host=dynamic
nat=no
canreivite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[016]
type=friend
context=my-phones
username=016
secret=016
host=dynamic
nat=no
canreivite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[017]
type=friend

context=my-phones
username=017
secret=017
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[018]
type=friend
context=my-phones
username=018
secret=018
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[019]
type=friend
context=my-phones
username=019
secret=019
host=dynamic

[020]
type=friend
context=my-phones
username=020
secret=020
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[021]
type=friend
context=my-phones

username=021
secret=021
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[022]
type=friend
context=my-phones
username=022
secret=022
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[023]
type=friend
context=my-phones
username=023
secret=023
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[024]
type=friend
context=my-phones
username=024
secret=024
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[025]
type=friend
context=my-phones
username=025
secret=025
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[026]
type=friend
context=my-phones
username=026
secret=026
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[027]
type=friend
context=my-phones
username=027
secret=027
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[028]
type=friend
context=my-phones
username=028
secret=028
host=dynamic

```
nat=no
canreinvite= yes      ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info
```

```
[029]
type=friend
context=my-phones
username=029
secret=029
host=dynamic
nat=no
canreinvite= yes      ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info
```

```
[030]
type=friend
context=my-phones
username=030
secret=030
host=dynamic
nat=no
canreinvite= yes      ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info
```

Extension.Conf

```
.....
[my-phones]
exten => 001, 1, Dial(SIP/001)
exten => 002, 1, Dial(SIP/002)
exten => 003, 1, Dial(SIP/003)
exten => 004, 1, Dial(SIP/004)
exten => 005, 1, Dial(SIP/005)
exten => 006, 1, Dial(SIP/006)
exten => 007, 1, Dial(SIP/007)
exten => 008, 1, Dial(SIP/008)
exten => 009, 1, Dial(SIP/009)
exten => 010, 1, Dial(SIP/010)
exten => 011, 1, Dial(SIP/011)
```

```
exten => 012, 1, Dial(SIP/012)
exten => 013, 1, Dial(SIP/013)
exten => 014, 1, Dial(SIP/014)
exten => 015, 1, Dial(SIP/015)
exten => 016, 1, Dial(SIP/016)
exten => 017, 1, Dial(SIP/017)
exten => 018, 1, Dial(SIP/018)
exten => 019, 1, Dial(SIP/019)
exten => 020, 1, Dial(SIP/020)
exten => 021, 1, Dial(SIP/021)
exten => 022, 1, Dial(SIP/022)
exten => 023, 1, Dial(SIP/023)
exten => 024, 1, Dial(SIP/024)
exten => 025, 1, Dial(SIP/025)
exten => 026, 1, Dial(SIP/026)
exten => 027, 1, Dial(SIP/027)
exten => 028, 1, Dial(SIP/028)
exten => 029, 1, Dial(SIP/029)
exten => 030, 1, Dial(SIP/030)
```

7.3. Konfigurasi OLSR

----- OLSR -----

```
config olsrd
option IpVersion '4'
```

```
config LoadPlugin
option library 'olsrd_arprefresh.so.0.1'
```

```
config LoadPlugin
option library 'olsrd_dyn_gw.so.0.5'
```

```
config LoadPlugin
option library 'olsrd_httpinfo.so.0.1'
option port '1978'
list Net '0.0.0.0 0.0.0.0'
```

```
config LoadPlugin
option library 'olsrd_nameservice.so.0.3'
```

```
config LoadPlugin
option library 'olsrd_txtinfo.so.0.1'
option accept '0.0.0.0'
```

```
config Interface
option ignore '0'
option interface 'mywifi'
```

```
option Mode 'mesh'
config InterfaceDefaults

config LoadPlugin
option library 'olsrd_bmf.so.1.7.0'
option ignore '1'

config LoadPlugin
option library 'olsrd_dot_draw.so.0.3'
option ignore '1'

config LoadPlugin
option library 'olsrd_dyn_gw_plain.so.0.4'
option ignore '1'

config LoadPlugin
option library 'olsrd_jsoninfo.so.0.0'
option ignore '1'

config LoadPlugin
option library 'olsrd_pgraph.so.1.1'
option ignore '1'

config LoadPlugin
option library 'olsrd_mdns.so.1.0.0'
option ignore '1'

config LoadPlugin
option library 'olsrd_p2pd.so.0.1.0'
option ignore '1'

config LoadPlugin
option library 'olsrd_quagga.so.0.2.2'
option ignore '1'

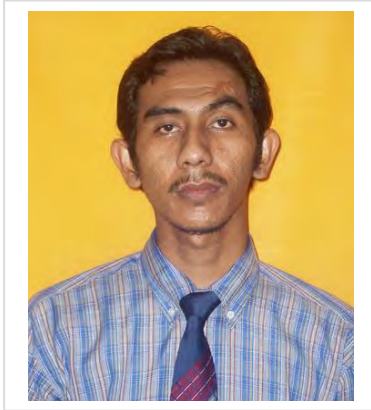
config LoadPlugin
option library 'olsrd_secure.so.0.6'
option ignore '1'

config LoadPlugin
option library 'olsrd_tas.so.0.1'
option ignore '1'

config LoadPlugin
option library 'olsrd_watchdog.so.0.1'
option ignore '1'
```

```
config Hna4
option netaddr '192.168.34.0'
option netmask '255.255.255.0'
```

RIWAYAT HIDUP



Edi Prihartono, lahir di Kota Surabaya JATIM, 28 Mei 1972, merupakan anak ketiga dari 4 bersaudara. Memulai pendidikan formal di SDN Menur Pumpungan V (1979-1985). Setelah lulus melanjutkan pendidikan di SMP EKA JAYA SURABAYA (1985-1988) dan melanjutkan pendidikan sekolah menengah atas di SMA MAHRDHKA SURABAYA (1988-1991).

tahun 1992 melanjutkan pendidikan kejenjang S1 di Universitas DR. SOETOMO SURABAYA (1992-1998), pada tahun 2000 sebagai tenaga pengajar di Universitas DR. SOETOMO. Pada tahun 2014 melanjutkan pendidikan S2 sebagai penerima beasiswa BPDN, mengambil program magister di Institut Teknologi Sepuluh Nopember dengan jurusan Teknik Elektro bidang keahlian TELEMATIKA (2014-2016).

Penulis bisa dihubungi melalui email edi081ta@yahoo.com

KATA PENGANTAR

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perkembangan telekomunikasi didunia semakin pesat dan cepat. Diawali dengan adanya komunikasi konvensional dengan jaringan PSTN, dan berkembang menjadi komunikasi modern dengan jaringan cellular. Sejalan dengan perkembangan teknologi komunikasi yang terus , Sarana komunikasi pun semakin meningkat jumlahnya dan beragam. Dengan meningkatnya jumlah sarana komunikasi maka, secara tidak langsung berakibat meningkat pula jumlah panggilan. Hal ini menjadikan kendala apabila besarnya jumlah panggilan tidak dapat dilayani oleh perangkat penerima panggilan komunikasi.

Bentuk alternatif komunikasi pada saat ini adalah komunikasi VoIP. Voip merupakan teknologi komunikasi suara jarak jauh yang dilewatkan pada media internet. Salah satu komponen penting dari voip adalah server voip. Server voip dapat dibangun dengan perangkat Personal Computer (PC). Dan juga bisa dibangun dengan perangkat AP/Router dengan system embeded system openWRT.

VoIP yang memanfaatkan Mobile Adhoc Network (MANet) sebagai jaringan untuk tranfer data, salah satu komponen utamanya adalah server voip. Dalam pembuatan server voip dengan menggunakan perangkat AP/Router, karena memberikan kelebihan mobilitas dan fleksibilitas dari sisi routing jaringan. Tetapi Server VoIP yang dibangun dengan menggunakan perangkat AP/Router ini, mempunyai keterbatasan. Salah satu keterbatasan tersebut adalah masalah kemampuan kapasitas dalam melayani jumlah panggilan oleh user/client yang terbatas atau sedikit (Asriadi, 2014). Sehingga hanya dapat diimplementasikan pada jaringan skala kecil. Keterbatasan tersebut disebabkan Semakin banyak panggilan pengguna yang masuk, sehingga kerja server akan semakin berat dan berakibat server tidak dapat lagi mampu melayani panggilan. Sedangkan jika

diimplementasikan pada jaringan skala besar dengan jumlah panggilan yang banyak akan terjadi penurunan pada kualitas layanan dikarenakan overhead.

Pada penelitian ini kami diusulkan suatu metode penanganan tranfer media data pada Server VoIP yaitu metode Native Bridging. Metode ini diharapkan mampu meningkatkan kapasitas jumlah panggilan yang dilayani oleh Server VoIP.

1.2 Rumusan Masalah

Dari latar belakang diatas didapat ditarik permasalahan, Voip dengan mengaplikasikan perangkat AP-Router sebagai VoIP Server, mempunyai kapasitas melayani jumlah panggilan yang terbatas atau sedikit, sehingga hanya cocok di aplikasikan pada jaringan skala kecil.

1.3 Batasan Masalah

Pada Penelitian ini, akan diberikan batasan masalah , yaitu :

1. Server VoIP menggunakan AP/Router TPLink -MR3020
2. VoIP codec yang digunakan jenis codec GSM
3. Client menggunakan Smartphone Android
4. Pengukuran jumlah panggilan dan kualitas voice
5. Skenario pengukuran secara real time, pada kondisi server dan client statis

1.4 Tujuan Penelitian

Tujuan penelitian ini, meningkatkan kapasitas panggilan VoIP yang mampu dilayani oleh Server VoIP berbasis embeded system dengan metode Native Bridging.

1.5 Manfaat Penelitian

1. Mengetahui peningkatan kapasitas jumlah maksimal panggilan yang dapat dilayani oleh server voip dengan metode Native Bridging.
2. Pada implementasi jaringan komunikasi, dapat ditentukan jumlah node. yang dibutuhkan untuk memenuhi jumlah pengguna pada suatu area.

1.6 Sistematika Penulisan

Sistematika penulisan pada penelitian ini terdiri dari beberapa bab, agar mudah dipahami, sebagai berikut :

BAB 1 : Pendahuluan

Pada bab pendahuluan berisi latar belakang, maksud dan tujuan penelitian, rumusan masalah, batasan masalah, manfaat dan metode penelitian yang digunakan, serta sistematika pembahasan.

BAB 2 : Tinjauan Pustaka

Pada bagian tinjauan pustaka berisikan teori pendukung yaitu tentang pengertian VoIP, MANet, protokol routing yang dipakai, jenis codec dan metode Native Bridging yang diujikan, serta aplikasi pendukung lainnya.

BAB 3 : Metoda Penelitian

Pada bab ini berisi tahapan tentang perencanaan penelitian meliputi penentuan hardware dan software yang akan digunakan, Instalasi software ke perangkat, konfigurasi protokol dan VoIP Codec, penentuan lokasi, skenario pengujian dan pengukuran untuk mendapatkan data.

BAB 4: Hasil dan Pembahasan

Pengimplementasian dari rancangan sistem dengan tahapan instalasi dan konfigurasi, pengujian dan pengukuran serta analisa data, dibahas pada bab ini. Yang meliputi data dan analisa panggilan sistem voip, kapasitas jumlah panggilan pernode, jarak jangkauan dan kapasitas panggilan dengan dua node

BAB 5 : Kesimpulan dan Saran

Pada bab ini berisikan kesimpulan dari hasil proses analisa dan saran untuk menyempurnakan penelitian ini.

BAB 2

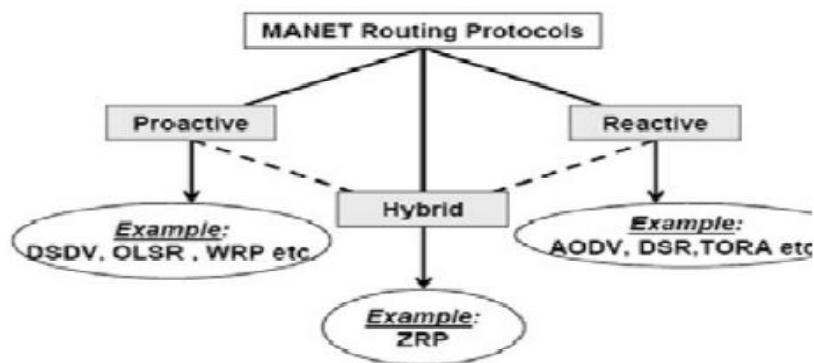
TINJAUAN PUSTAKA

2.1 Manet Adhoc Network(MANET)

Mobile Adhoc Network atau yang lebih dikenal dengan MANET merupakan jaringan wireless yang memiliki node bergerak. Ciri khusus yang ada pada jaringan ini adalah setiap node yang dalam jaringan dapat melakukan proses routing dan pengiriman data. Node bertanggung jawab atas proses routing discovery dan routing maintenance untuk setiap jalur pengiriman data ke node destinasi. Hal ini dikarenakan setiap node yang berada dalam jaringan selalu bergerak. Adanya pergerakan yang random dari setiap node menyebabkan perubahan topologi pada jaringan MANET secara dynamic. Hal ini menyebabkan perubahan yang dynamic pada konektivitas antar node, yang saling bertukar data. Oleh karena itu diperlukan adanya routing protokol yang dapat mengcover kebutuhan jaringan untuk dapat memberikan jalur routing secara optimal (Samrat Ganguly, 2006).

2.2 Routing Protokol Pada Manet

Adanya perubahan yang dinamic pada konektifitas antar node pada jaringan MANet maka dibutuhkan routing protokol yang dapat menjaga koneksifitas yang handal pada jaringan MANet untuk proses pertukaran informasi.



Gambar .2.1 Klasifikasi protokol pada MANet

Pada gambar 2.1 menunjukkan klasifikasi routing protokol pada MANet yang dibedakan menjadi Protokol Reaktif, Proaktif dan Hybrid (Hrituparna Paul A, 2013).

2.2.1 Routing Protokol Reaktif

Jenis routing protokol ini, dikenal sebagai routing yang bekerja bila ada permintaan atau routing protokol yang bersifat reaktif. Pencarian rute terpendek untuk transmisi data baru akan dilakukan saat ada data yang akan dikirim. Akibat dari mekanisme tersebut adalah terjadinya delay yang sedikit lama saat akan melakukan pengiriman data. Protokol AODV, DSR dan TORA merupakan reaktif routing protokol.

2.2.2 Routing Protokol Proaktif

Routing protokol proaktif juga disebut sebagai *table driven* routing. Pada jenis 1 routing protokol ini, tiap node diharuskan mempunyai tabel untuk menyimpan informasi daftar terbaru tujuan dan rute, distribusi tabel routing seluruh jaringan. Protokol ini selalu berusaha untuk menjaga konsisten dan memperbarui informasi routing pada setiap node (Hrituparna Paul A, 2013). Protokol routing proaktif menggunakan algoritma routing link-state, yang sering dipenuhi informasi dari node yang terhubung dengannya. Kelemahan utama proaktif routing protokol adalah bahwa semua node dalam jaringan selalu menjaga update tabel. Protokol OLSR, DSDV dan WRP adalah protokol routing proaktif (Saputra, 2013)

2.2.3 Routing Protokol Hybrid

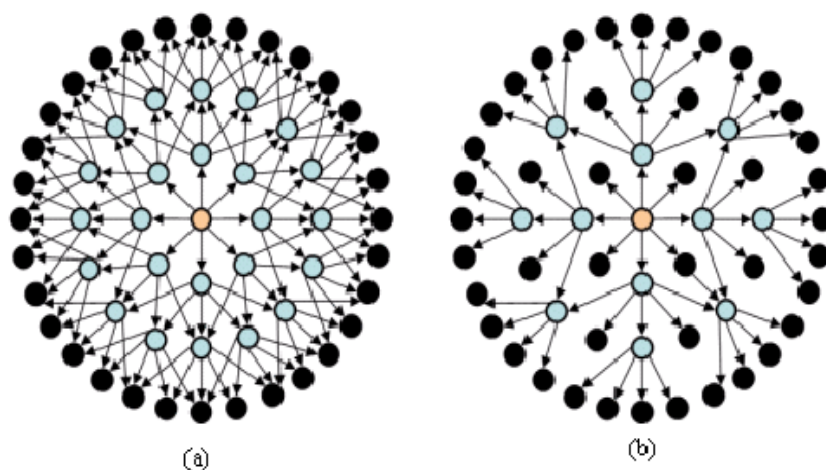
Hybrid routing protokol adalah routing protokol yang dibuat dengan mengambil keunggulan dari routing protokol reaktif dan proaktif, dengan kata lain menggabungkan keunggulan dari mekanisme kedua routing protokol tersebut. Dalam implementasinya, routing protokol jenis ini membutuhkan peralatan tambahan seperti GPS. Protokol ZRP adalah contoh dari routing protokol jenis ini.

2.2.4 Routing Protokol OLSR

Optimal Link State Routing (OLSR) Protokol adalah protokol routing proaktif dimana pada jenis routing ini jalur rute selalu tersedia bila

diperlukan. OLSR merupakan versi optimasi dari protokol link state murni, di mana perubahan topologi menyebabkan luapan dari informasi topologi ke semua host yang tersedia di jaringan, gambar 22.a. OLSR dapat mengoptimalkan reaktifitas Perubahan topologi dengan mengurangi waktu maksimum interval untuk transmisi pesan control berkala. Selain itu, sebgaiian OLSR terus mempertahankan rute untuk semua tujuan dalam jaringan, protocol ini bermanfaat bagi pola lalu lintas di mana subset node besar berkomunikasi dengan bagian node besar lainnya, dan pada saat pasangan berubah dari waktu ke waktu.

Protokol OLSR tidak cocok diaplikasikan pada transmisi paket data dengan keterlambatan lama dan bekerja dengan baik pada jaringan padat, dimana sebagian besar komunikasi terkonsentrasi antara sejumlah node besar. Sehingga terjadi adanya luapan informasi pada node utama. Untuk mengurangi jumlah *overhead* dalam jaringan yaitu, dengan menggunakan teknik *Multi Point Relays* (MPR). Tujuan utama dari MPR yaitu mengurangi luapan dengan cara memilih beberapa *node* untuk bertindak sebagai MPR, sehingga hanya *node-node* MPR saja yang dapat meneruskan paket kontrol yang diterima gambar 22.b Kinerja protokol OLSR yang paling efisien dalam jaringan padat. (Hriptuarna Paul A, 2013).



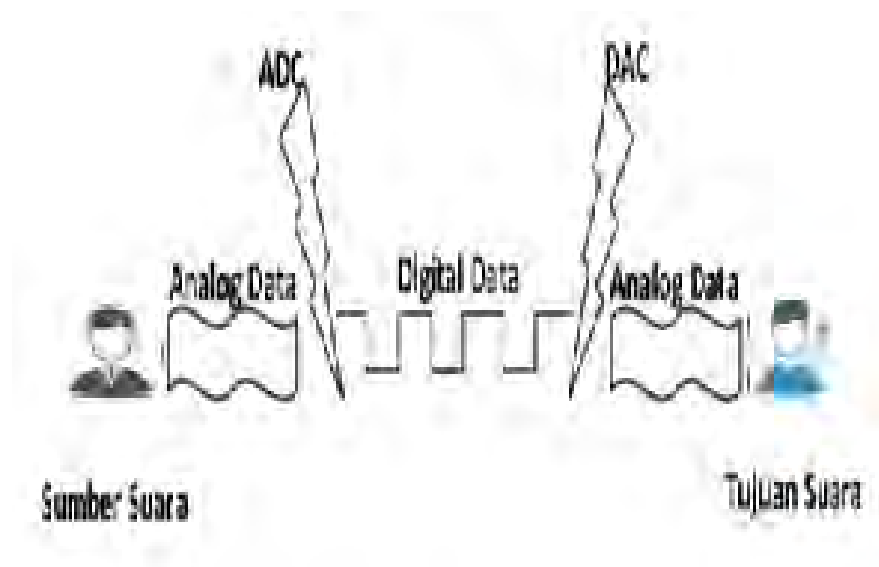
Gambar 2.2 Perbandingan Sistem broadcasting (a) Link state broadcast
(b) MPR broadcast

2.2.5 OLSR On Daemon

OLSR on Daemon (OLRSd) adalah bentuk implementasi dari OLSR routing protokol untuk MANET Olsrd dirancang untuk memudahkan penggunaan dan pemantauan dari penggunaan OLSR pada jaringan MANET serta penambahan plugin tertentu untuk mengefektifkan jaringan yang dibuat (Asriadi, 2014), (Hrituparna paul A ,2013). Olsrd dapat berjalan diberbagai platform seperti PC dekstop, single board computer, komputer berbasis ARM serta embedded device seperti router dan ponsel. Namun kebanyakan dari itu berjalan dengan sistem operasi yang berbasiskan GNU/Linux.

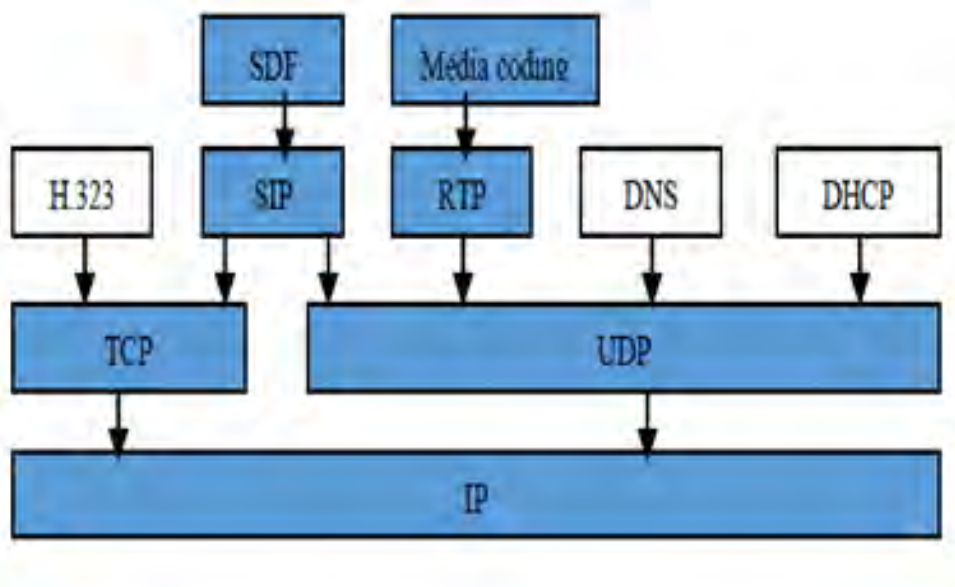
2.3 Voice Internet Protokol (VoIP)

VoIP (*Voice over Internet Protocol*) adalah teknologi yang mampu melewati trafik suara, *video* dan data yang berbentuk paket melalui jaringan IP. Jaringan IP adalah jaringan komunikasi data yang berbasis *packet-switch*, sehingga komunikasi telepon dapat dijalankan pada jaringan IP atau internet maupun intranet (Hrituparna paul A, 2013). Pengiriman suara pada komunikasi VoIP dengan cara merubah suara *analog* menjadi data-data *digital*. Perubahan ini dilakukan dengan menggunakan *Analog to Digital Converter* (ADC/DAC) ditunjukkan pada Gambar 2.3. Data *digital* ini kemudian dikirimkan kepada penerima melalui jaringan Internet. Ketika data *digital* tersebut diterima, maka data *digital* tersebut akan di konversi kembali menjadi suara *analog* yang dapat didengarkan oleh penerima.



Gambar 2.3 Cara Kerja VoIP

Selain ADC/DAC untuk merubah data analog ke data digital atau sebaliknya. Pada VoIP Juga bekerja beberapa protokol pendukung, berdasarkan fungsinya protokol pada VoIP dapat dibedakan menjadi 2 yaitu protokol persinyalan dan media transfer. Protokol rotkol VoIP tersebut ditunjukkan pada gambar 2.4 dibawah ini.



Gambar 2.4 Hirarki Protokol VoIP

2.3.1 Protokol Persinyalan .

Berfungsi untuk membangun, menjaga suatu sesi komunikasi yang sedang berlangsung dan memutus suatu koneksi. Berikut beberapa jenis protokol persinyalan.

a. **Session Initiation Protocol (SIP)**

Standarisasi protokol komunikasi pada teknologi VoIP adalah SIP (*Session Initiation rotocol*) dan H.323. Tetapi dalam perkembangannya SIP lebih banyak digunakan oleh pengembang teknologi VoIP (Asriadi,2014). SIP merupakan sebuah session layer protokol yang digunakan untuk membentuk, memodifikasi, dan menterminasi sebuah sesi multimedia. Seperti layaknya HTTP, SIP merupakan client-server protokol yang menggunakan

model transaksi request dan response. SIP juga telah banyak diadopsi oleh banyak software dan hardware dipasaran yang menjadikannya lebih mudah dalam implementasi (Istas P,2013)

b. Inter Asterisk Exchange (IAX)

IAX merupakan signaling protokol yang dibuat sebagai alternatif dari SIP. protokol tersebut bisa digunakan untuk komunikasi multimedia baik itu suara maupun video, namun terbatas hanya untuk komunikasi berbasis IP. Selain itu IAX adalah media dan signaling protocol point to point yang bekerja berbasiskan signal multiplexing dan multiplestreams melalui UDP diantara dua user pada port 4569.

Dibandingkan dengan SIP, IAX mempunyai beberapa kelebihan diantaranya konsumsi bandwidth yang lebih rendah, mendukung NAT transparan. Tetapi implementasi protokol ini ke peralatan-peralatan pendukung belum banyak dipasaran. Berbeda dengan SIP yang telah banyak diadopsi oleh banyak software dan hardware dipasaran yang menjadikannya lebih mudah dalam implementasi.

2.3.2 Protokol Media Transfer

Berfungsi untuk mengatur komunikasi pada saat transfer data (baik *voice*, *video*, maupun data) secara *realtime*. Berikut adalah protokol-protokol media transfer :

a. TPC/IP (Transfer Control Protocol/Internet Protocol)

TPC/IP merupakan protokol yang digunakan pada jaringan Internet. Dan terdiri dari dua bagian besar, yaitu TCP dan IP. Protokol TCP merupakan protokol yang connection-oriented yang artinya menjaga reliabilitas hubungan komunikasi end-to-end . Konsep dasar kerja TCP adalah mengirim dan menerima segment-segment informasi, dengan panjang data bervariasi pada suatu datagram internet.

TCP menjamin reliabilitas hubungan komunikasi karena melakukan perbaikan terhadap data yang rusak, hilang atau kesalahan kirim. Hal ini dilakukan dengan memberikan nomor urut pada setiap data

yang dikirimkan dan membutuhkan sinyal jawaban positif dari penerima berupa sinyal ACK (Acknowledgment). Jika sinyal ACK ini tidak diterima pada interval pada waktu tertentu, maka data akan dikirimkan kembali. Pada sisi penerima, nomor urut tadi berguna untuk mencegah kesalahan urutan data dan duplikasi data.

TCP juga memiliki mekanisme flow control dengan cara mencantumkan informasi dalam sinyal ACK mengenai batas jumlah oktet, data yang masih boleh ditransmisikan pada setiap segment yang diterima dengan sukses. Dalam hubungan dengan VoIP, TCP digunakan pada saat signaling, TCP digunakan untuk menjamin setup suatu call pada sesi signaling. TCP tidak digunakan dalam pengiriman data suara pada VoIP karena pada suatu komunikasi VoIP penanganan paket yang mengalami keterlambatan lebih penting daripada penanganan paket yang hilang (Dimas Lazuardi, 2008).

b. UDP (*User Datagram Protocol*)

UDP Merupakan salah satu protokol utama di atas IP dan merupakan protokol media transfer yang lebih sederhana, dibandingkan dengan TCP. UDP digunakan untuk situasi yang tidak mementingkan mekanisme reliabilitas. Artinya pada protokol UDP ini komunikasi akan tetap berlangsung tanpa memperdulikan koneksi antara sumber dan tujuan. Protokol UDP juga tidak melakukan perbaikan terhadap paket yang rusak atau hilang pada saat pengiriman paket suara berlangsung. Header UDP hanya berisi empat field yaitu source port, destination port, length dan UDP checksum yang fungsinya hampir sama dengan TCP, namun fasilitas checksum audio streaming yang secara terus menerus pada UDP bersifat opsional.

UDP pada VoIP digunakan untuk mengirimkan audio streaming yang secara terus menerus. Protokol ini lebih mementingkan kecepatan pengiriman paket data agar tiba di tujuan tanpa memperhatikan adanya paket yang hilang walaupun mencapai 50% dari jumlah paket yang dikirimkan. Karena UDP mampu mengirimkan data streaming dengan

cepat, sehingga dalam teknologi VoIP, UDP merupakan salah satu protokol penting yang digunakan sebagai header pada pengiriman data.

c. RTP (*Real Time Protocol*)

RTP adalah protokol yang didesain untuk membawa paket audio atau video melalui jaringan berbasis IP. Paket pada RTP memuat potongan paket-paket audio ataupun video yang ditransmisikan dengan bantuan protokol UDP. Karena bekerja pada UDP maka jika terdapat paket RTP yang tidak diterima dengan baik atau hilang saat transmisi, maka RTP tidak akan melakukan pengiriman paket ulang.

d. SRTP (*Secure Real Time Protocol*)

Protokol SRTP sebagai penyedia fitur enkripsi pada profil RTP. Fitur enkripsi ini diperuntukkan sebagai sistem keamanan data, dengan otentikasi dan integritas pesan serta perlindungan terhadap playback data RTP dalam aplikasi unicast dan multicast. Dengan metode algoritma AES (*Advanced Encryption Standard*) untuk mengenkripsi data dalam proses pengiriman.

Pada aplikasi, SRTP terdiri dari 3 mode, yaitu Segmented AES di f8-mode, Integer Counter dan null cipher. Pada Mode Null Cipher, pengiriman data tidak terlindungi oleh adanya enkripsi. SRTP pada kenyataannya hanya mengenkripsi payload (audio dan video) untuk kerahasiaan.

2.3.3 Coder dan Decoder (Codec) VoIP

Codec merupakan sebuah proses mengubah data suara (analog) ke dalam bentuk data digital yang selanjutnya dimampatkan (Kompresi), kemudian ditransmisikan dan dikembalikan lagi ke bentuk sinyal suara ketika sampai ke tujuan secara *realtime* melalui *jaringan Internet Protokol* (Asriadi, 2014). Sehingga peranan codec pada komunikasi VoIP sangat penting sekali. Karena besar kecilnya hasil proses kompresi data oleh codec dapat menghemat bandwidth dan mempengaruhi kualitas suara pada komunikasi VoIP.

Codec tersedia dalam bentuk *open source* dan *non-open source*. Tabel 2.1 menunjukkan daftar dari beberapa jenis codec .

Tabel 2.1 Jenis Codec

Codec	Bitrate (kb/s)	Frame (ms)	Bits per frame	Algorithmic delay ^a (ms)	Codec delay ^b (ms)	Compression type	Complexity (MIPS) ^c	MOS
<i>Narrowband codecs</i>								
G.711	64	0.125	8	0.125	0.25	PCM	≪1	4.1 ^d
G.723.1	6.3	30	189	37.5	67.5	MP-MLQ	≪18	3.8
G.723.1	5.3	30	159	37.5	67.5	ACELP	≪18	3.6
G.726	16	0.125	2	0.125	0.25	ADPCM	≈1	-
G.726	24	0.125	3	0.125	0.25	ADPCM	≈1	3.5
G.726	32	0.125	4	0.125	0.25	ADPCM	≈1	4.1
G.728	16	0.625	10	0.625	1.25	LD-CELP	≈30	3.61
G.729	8	10	80	15	25	CS-ACELP	≪20	3.92
G.729A	8	10	80	15	25	CS-ACELP	≪11	3.7
G.729D	6.4	10	64	15	25	CS-ACELP	<20	3.8
G.729E	11.8	10	118	15	25	CS-ACELP LPC	<30	4
GSM-FR	13	20	260	20	40	RPE-LTP	≈4.5	3.6
GSM-HR	5.6	20	112	24.4	44.4	VSELP	≈30	3.5
GSM-EFR	12.2	20	244	20	40	ACELP	≈20	4.1
AMR-NB	4.75–12.2	20	95–244	25	45	ACELP	15–20	3.5–4.1
iLBC	13.33	30	400	40	60	LPC	18	3.8
iLBC	15.2	20	304	25	40	LPC	15	3.9
Speex (NB)	2.15–24.6	20	43–492	30	50	CELP	8–25	2.8–4.2
BV16	16	5	80	5	10	TSNFC	12	4
<i>Broadband codecs</i>								
G.722	48, 56, 64	0.0625	3–4	1.5	1.5625	SB-ADPCM	5	~4.1
G.722.1	24, 32	20	480, 640	40	60	MLT	<15	~4
AMR-WB (G.722.2)	6.6–23.85	20	132–477	25	45	ACELP	≈38	Various
Speex (WB)	4–44.2	20	80–884	34	50	CELP	8–25	Various
iSAC	Variable	Adaptive	Adaptive-variable	Frame + 3 ms	Adaptive	Transform	6–10	Various ^e
	10–32	30–60 ms			63–123	coding		
BV32	32	5	160	5	10	TSNFC	17.5	~4.1

Jenis codec yang banyak digunakan pada komunikasi voip adalah codec GSM (Global System Mobile). Codec ini merupakan salah satu jenis codec VoIP yang mempunyai kinerja sangat baik untuk kompresi audio (OpenWRT, 2012) .

Codec GSM merupakan jenis pengkodean suara yang melakukan pengompresian paket suara dengan teknik RPE-LTP (*Regular Pulsa Excitation Long Term prediction*). Codec ini menggunakan sampel sebanyak 16.000 kali/detik dengan bit rate 8 bit/detik sehingga menghasilkan laju rate sebesar 13.000 bit/detik. RPE-LTP mengkonversi sinyal analog ke bentuk digital dengan

melakukan sampling sinyal analog tersebut 8000 kali per detik dan dikodekan dalam angka. Jarak antar sampel adalah 20,000 μ s. Sinyal tersampel lalu dikonversikan ke bentuk diskrit. Sinyal diskrit ini direpresentasikan dengan kode yang disesuaikan dengan amplitude dari sinyal sampel. Format RPE-LTP menggunakan 260 bit untuk pengkodeannya. Laju transmisi diperoleh dengan mengalikan 8000 sampel per detik dengan 260 bit per sampel, menghasilkan 13.000 bit per detik. Bit rate 13 Kbps ini merupakan standar transmisi untuk suatu kanal telepon digital

2.3.4 *Quality Of Service (QOS)*

Suatu metode pengukuran kemampuan dalam menyediakan tingkat layanan untuk transmisi data pada suatu jaringan. Adapun parameter QOS adalah :

1. *Throughput*

Throughput menunjukkan jumlah *bit* yang diterima dengan sukses per detik melalui sebuah sistem atau media komunikasi dalam selang waktu tertentu yang pada umumnya dilihat dalam satuan *bits/sec*

2. *Delay*

Delay menunjukkan waktu tunda yang terjadi pada suatu data ketika ditransmisikan dari *transmitter* menuju *receiver*. Waktu tunda merupakan suatu permasalahan yang harus diperhitungkan karena kualitas suara bagus tidaknya tergantung dari waktu tunda. Besarnya waktu tunda maksimum yang direkomendasikan oleh ITU-T G.114 untuk aplikasi suara adalah 150 ms, sedangkan waktu tunda maksimum dengan kualitas suara yang masih dapat diterima pengguna adalah 250 ms.

Tabel 2.2 Kualitas Nilai Delay

Nilai Delay	Kualitas
0 - 150 ms	Baik
150 - 250 ms	Cukup
> 250 ms	Buruk

3. *Jitter*

Jitter merupakan variasi delay yang terjadi karena adanya selisih waktu antar kedatangan paket yang di penerima. Untuk mengatasi jitter, paket data yang datang dikumpulkan terlebih dulu dalam buffer selama waktu yang telah ditentukan sampai paket dapat diterima keseluruhan pada sisi penerima, dengan urutan yang benar. Adanya buffer tersebut akan mempengaruhi waktu tunda total sistem, akibat adanya tambahan proses untuk mengompensasi terjadinya jitter. Pada tabel 2.3 menunjukkan, jitter berkategori baik jika mempunyai 0 – 20 ms dan masih dalam batas toleransi apabila bernilai 20 – 50 ms, sedangkan kategori buruk jika nilai jitter lebih dari 50 ms

Tabel 2.3 Kualitas Nilai Jitter

Nilai Jitter	Kualitas
0-20 ms	Baik
20-50 ms	Cukup
>50 ms	Buruk

4. *Paket Loss*

Packet Loss adalah hilangnya paket data yang dikirim ketika terjadi Peak load dan congestion (kemacetan transmisi paket akibat padatnya traffic yang harus dilayani) dalam batas waktu tertentu, maka frame (gabungan data payload dan header yang di transmisikan) suara akan dibuang, sebagaimana perlakuan terhadap frame data lainnya pada jaringan berbasis IP, sehingga ada kemungkinan terjadinya hilang paket pada saat transmisi. Tingkat paket hilang ini tentunya akan mempengaruhi kualitas layanan. Pada tabel 2.4, menunjukkan packet loss berkualitas baik apabila 0 – 1,5%, dan tidak dapat di terima apabila > 1,5% dalam aplikasi suara.

Tabel 2.4 Kualitas Nilai Paket Loss

Packet Loss	Kualitas
0-0,5 %	Sangat baik
0,5-1,5 %	Baik
>1,5 %	Buruk

2.4 Operating Sistem untuk Router Wireless

OpenWRT merupakan salah satu distribusi linux yang didesain untuk perangkat *embedded*. Salah satu penerapan sistem operasi OpenWRT adalah pada wireless router.

OpenWRT diciptakan semula untuk mendukung *wireless router* Linksys WRT54G, tapi perkembangan OpenWRT selanjutnya dapat digunakan pada beberapa manufaktur wireless router yang lain seperti TP-Link, D-Link, ASUS, DELL dan lain-lain. OpenWRT menyediakan *file system* dengan modul-modul dari aplikasi terpisah yang dapat diimplementasikan pada OpenVoice (Asriadi,2014). *Modul-modul* tersebut di antaranya uClibc, busybox, shell interpreter, abstraksi perangkat keras dan juga paket manager Aplikasi Voip

2.4.1 Sistem pada VoIP Server

Asterisk adalah framework yang bersifat open source yang digunakan untuk membangun sebuah jaringan komunikasi. Asterisk mencakup banyak fitur yang tersedia dalam sistem PBX seperti voice mail, teleconference, VoIP, dan distribusi panggilan otomatis. Asterisk mendukung berbagai fitur multimedia seperti Voice over IP protokol, dengan menggunakan protokol Session Initiation Protocol (SIP), Media Gateway Control Protocol (MGCP), dan H.323. Asterisk dapat berperan baik sebagai registra atau sebagai gateway antara VoIP dan PSTN. Kelebihan asterisk dengan ukuran sistem yang kecil, sehingga aplikasi ini memungkinkan aplikasi ini berjalan pada embeded system seperti OpenVoice (OpenWRT, 2014). dan menjadikan router biasa menjadi sebuah VoIP Server

Asterisk adalah salah satu paket modul yang diinstal pada OpenWRT dan akan berada pada bagian User Program. Paket-paket yang membutuhkan akses ke hardware akan melewati Linux Kernel dan kemudian akan diteruskan ke hardware

tujuan. Asterisk itu sendiri memberikan kesederhanaan bagi pengguna untuk meningkatkan layanan telepon sendiri dengan kustomisasi yang fleksibel oleh pengguna.

2.4.2 TP Link MR-3020

Perangkat TP Link MR3020 adalah sejenis Router/Akses Point wireless. Yang mempunyai spesifikasi sebagai berikut , berdimensi 2.9 x 2.6 x 0.9 inchi (74 x 67 x22 mm), karena ukurannya yang sangat kecil sehingga mudah untuk dibawa. Berdaya 5VDC/1.0A konsumsi listrik sangat kecil. Mempunyai frekuensi 2.4-2.4835GHz. Didukung System-On-Chip(SoC) Atheros AR9331, CPU Speed 400MHz, RAM 32 MB, Flash Memory 4 MB Wireless Chip Atheros AR9331 1x1:2, perangkat ini selain mempunyai spesifikasi diatas, yang lebih penting type MR3020 ini dapat ditanam aplikasi fireware OpenWRT, asteriks dan dapat ditambahkan external storage. Sehingga perangkat router jenis ini sangat cocok dan sesuai kebutuhan, untuk dijadikan sebagi Server VoIP pada penelitian ini (Asriadi,2014).

Spesifikasi lengkap router MR-3020 yang akan dipakai dalam penelitian ini dapat dilihat pada Tabel 2.5.

Tabel 2.5 Spesifikasi Router MR-3020

No	Spesifikasi	Ket.
1	Arsitektur	MIPS 24Kc V7.4
2	Vendor	Atheros
3	Bootloader	U-boot 1.1.3
4	System-On-Chip	Atheros AR9331
5	CPU Speed	400MHz
6	RAM Memory	32 MB SDRAM
7	Flash Memory	4 MB
8	Flash Chip	Windbond W9425G6JH
9	Wired Network	2x Ethernet 100 Mbps (switched)
10	Wireless Chip	Atheros AR9331 1x1:2
11	Wireless antennas	2x printed on-board
12	USB	1 x USB 2.0 host

13	Serial Port	Yes (TTL pins)
----	-------------	----------------

2.4.3 Sistem pada VoIP Client

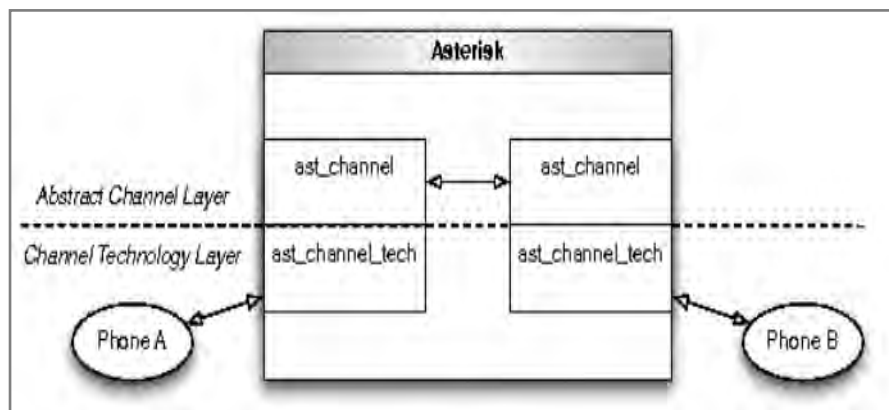
Pada sisi client sistem operasi yang digunakan adalah Android untuk handphone, sedangkan pada PC menggunakan sistem operasi windows juga bisa digunakan sistem operasi opensource.

Pada client yang menggunakan sistem android, sedangkan untuk aplikasi komunikasinya menggunakan softphone. Di mana softphone memiliki jenis yang beragam baik dari kemampuan dan lisensi. Saat ini banyak Softphone yang disebarkan dengan lisensi gratis. Bahkan ada yang menyediakan lisensi software gratis sekaligus untuk layanan jaringan VoIP-nya. Aplikasi softphone tersebut diantaranya adalah Zoiper. Zoiper merupakan aplikasi yang dapat digunakan untuk komunikasi suara melalui jaringan internet atau yang biasa kita sebut VoIP.

Sedangkan untuk PC yang menggunakan sistem operasi windows atau opensource, aplikasi yang dapat digunakan antara lain X-Lite, IAX-Lite, dan MyPhone. X-Lite merupakan aplikasi untuk VoIP yang berjalan melalui protokol SIP. Selain suara, X-Lite juga bisa digunakan untuk saling berkirim text dan video.

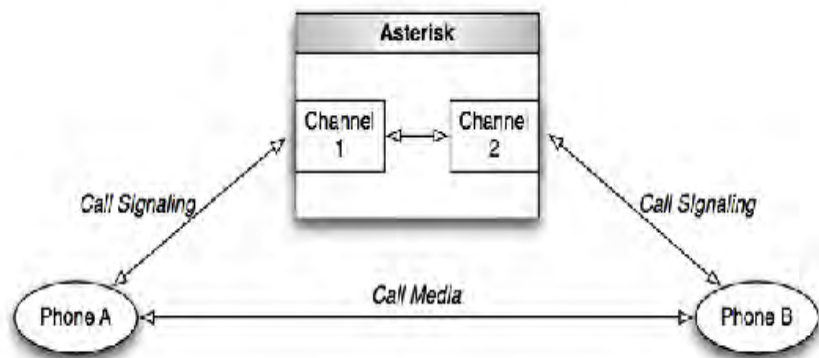
2.5 Bridging

Pada umumnya proses komunikasi pada voip menggunakan model saluran *Generic Bridging*. Saluran Generic Bridging adalah tindakan yang menghubungkan saluran secara bersama-sama untuk tujuan melewati media antar client. Dapat dilihat pada Gambar 2.5 di mana dua kaki panggilan diwakili oleh dua saluran ketika terhubung ke saluran inti Asterisk. **Xxxx** Di mana aliran media yang paling sering yaitu audio. Namun, ada juga mungkin video atau aliran teks dalam panggilan. Bahkan dalam kasus di mana ada lebih dari satu media streaming baik audio dan video, masih ditangani oleh satu saluran untuk setiap akhir panggilan di Asterisk. (Anonymous Asterisk, 2014)



Gambar 2.5 Saluran Bridging

Native Bridging: adalah metode penanganan tranfer media pada voip dengan cara bypass . Gambar 2.6, menunjukkan dimana setelah komunikasi(call signal) tersambung antar client (Phone A dan Phone B), paket suara tidak dilewatkan ke inti server (asterisk) lagi, tetapi paket suara (Call media) dari kedua client langsung dipertukarkan antar client (endpoint). Sehingga dapat menaikkan jumlah client yang mampu dilayani secara bersamaan. Tapi resikonya ketika kedua client tidak suport codec yang sama, komunikasinya tidak akan tersambung. (Effan Najwaini,2014)



Gambar 2.6 Skema Native Bridging

2.6 Aplikasi Pengukuran

Untuk pengukuran dan menganalisa proses kerja jaringan, bisa digunakan beberapa software berfungsi, diantara yaitu: *Wireshark* merupakan perangkat lunak yang spesifik untuk melakukan analisa paket data pada jaringan secara real

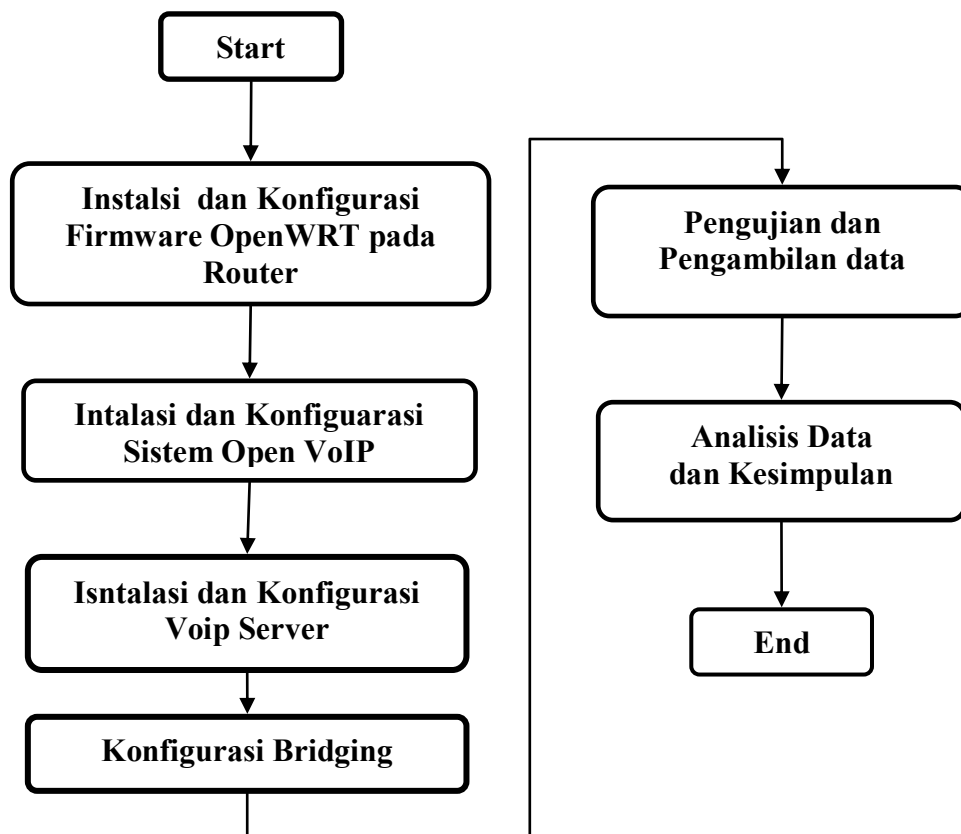
time dan menampilkan hasil analisa paket data tersebut dalam format yang dipahami oleh pengguna. Wireshark dapat melakukan paket filtering, paket color coding, dan fitur-fitur lain yang dapat mengizinkan untuk melihat detail network traffic dan inspeksi paket data secara individu. Wireshark dapat menganalisis paket data secara real time. Artinya, aplikasi wireshark akan mengawasi semua paket data yang keluar masuk melalui antarmuka yang telah ditentukan dan selanjutnya akan menampilkan hasil paket datanya. Wireshark dapat melakukan analisis terhadap beberapa protokol paket data jaringan.

BAB 3

METODOLOGI PENELITIAN

3.1 Rancangan Penelitian

Dalam perancangan penelitian ini, akan dilakukan tahapan-tahapan untuk membangun sistem yang akan buat, tahapan tersebut terlihat pada gambar 3.1. Diagram Flowchart Sistem.



Gambar 3.1 Digram Flowchat Sistem

3.1.1 Tahapan Instalasi Sistem OpenWRT pada Router

Tahapan awal untuk membangun sebuah server voip, adalah melakukan instalasi sebuah firmware OpenWRT Barrier Beker 1.4 yang suport dengan

perangkat AP/router TP Link MR3020 yang akan dijadikan sebagai VoIP Server, di mana firmware tersebut akan terembedded pada perangkat. Firmware pada embedded sistem berbasiskan OpenWRT sebagai frameworknya, asterisk sebagai softswitch dan digunakan dalam membuat SIP server untuk layanan suara serta prosody untuk layanan pesan. Digunakan firmware khusus ini bertujuan untuk mendapatkan firmware yang sesuai dengan kebutuhan aplikasi. Karena dalam firmware tersebut telah built-in dengan modul-modul yang diinginkan seperti asterisk, prosody, routing protokol serta mendukung beberapa jenis codec, sehingga dapat dipakai untuk komunikasi pesan dan suara serta dapat digunakan untuk meningkatkan quality of service dari sistem.

3.1.2 Tahapan Instalasi Open VoIP Sistem

Pada tahapan ini akan dilakukan beberapa instalasi dan konfigurasi adalah :

a. Konfigurasi Exroot

Keterbatasan spesifikasi hardware terutama pada internal storage yang minim kapasitas, bisa kita atasi dengan cara exroot. Dimana konfigurasi exroot ini merupakan salah satu cara untuk menambah internal storage pada AP/router yang sudah embedded openwrt, dengan menggunakan perangkat external storage berupa flashdisk.

b. Konfigurasi Server VoIP

Pada tahapan ini akan dilakukan konfigurasi asterisk yang akan menjadi SIP server dalam melayani pengguna yang terhubung dengan system ini.

c. Konfigurasi Native Bridging

Pada tahapan ini, akan dilakukan konfigurasi pada file sip.conf dan extension.conf. Konfigurasi ini menambahkan beberapa variabel khusus dan merubah nilai standart. sehingga akan mengaktifkan mode bridging menjadi Native Bridging.

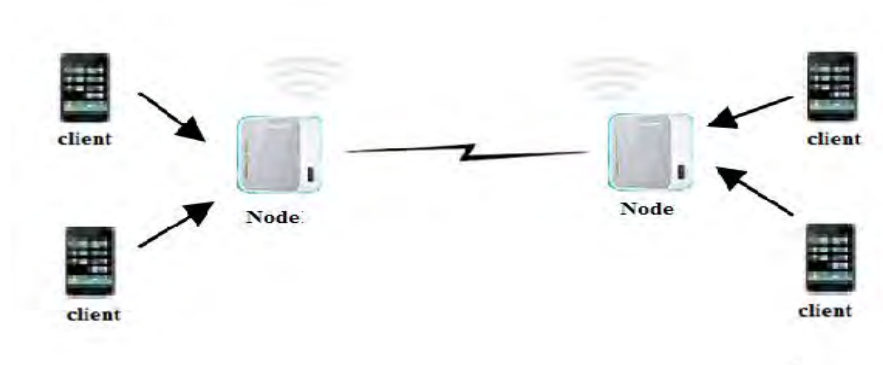
3.2 Gambaran Umum Sistem

Pada desain rancangan sistem, server voip yang dibuat dengan menggunakan firmware yang berbasis OpenWRT Barrier Beker 1.4 sebagai

frameworknya, di embededkan pada perangkat router wireless TP Link MR-3020.

Sebagai softswitch untuk membangun SIP Server (Server VoIP) akan digunakan aplikasi open source asterisk, yang berfungsi untuk melayani suara pada jaringan wireless voip. Di mana akan mengkonfigurasi asterisk pada mode native bridging untuk meningkatkan kapasitas layanan panggilan pada Server VoIP .

Pada sisi client, menggunakan smartphone sebagai perangkat komunikasi, dengan sistem operasi android. Dan telah terinstal dengan aplikasi softphone Zoiper sebagai media untuk berkomunikasi di client, yang terhubung secara wireless pada frekuensi ISM band 2,4 GHz kepada node-node yang ada. Arsitektur jaringan pada saat pengimplementasian prototype ini pada jalur akses dapat diilustrasikan seperti pada gambar 3.2



Gambar 3.2 Arsitektur Jaringan VoiP

3.3 Skenario Pengujian

Pada tahap ini akan dilakukan pengujian dan pengukuran terhadap server VoIP dengan menggunakan Router MR3020 yang telah embededkan firmware dan konfigurasi routing protokol serta Codec yang digunakan adalah jenis codec GSM(Global S Mobile), mendukung penerapan VoIP dalam sistem ini, menjadi sebuah Server VoIP.

Pengujian ini dibagi dalam beberapa skenario. Hal ini dilakukan agar mendapatkan hasil pengukuran yang mendekati pada saat benar-benar

diimplementasikan. Berikut penjelasan mengenai skenario serta kondisi yang diterapkan saat pengukuran dilakukan :

1. Kondisi smarphone yang dipakai tidak dalam keadaan terpasang simcard operator, untuk mensimulasikan tidak adanya layanan operator yang dipakai. Begitu pula dengan kondisi jaringan yang dipakai, tidak ada sama sekali layanan internet yang dipakai sehingga sistemnya benar-benar bekerja dalam keadaan offline
2. Semua pengujian dan pengukuran dilakukan secara real time, tidak menggunakan aplikasi simulator, artinya pengujian dilakukan dengan melakukan pemanggilan secara langsung.
- 3 Lokasi yang digunakan dalam pengukuran adalah tanah lapang untuk mengetahui performance dari sistem pada saat keadaan sinyal bebas halangan
4. Skenario pengujian dalam kondisi statis : dimana semua node dan Client yang ada di dalam sistem dalam keadaan diam dan tak bergerak.

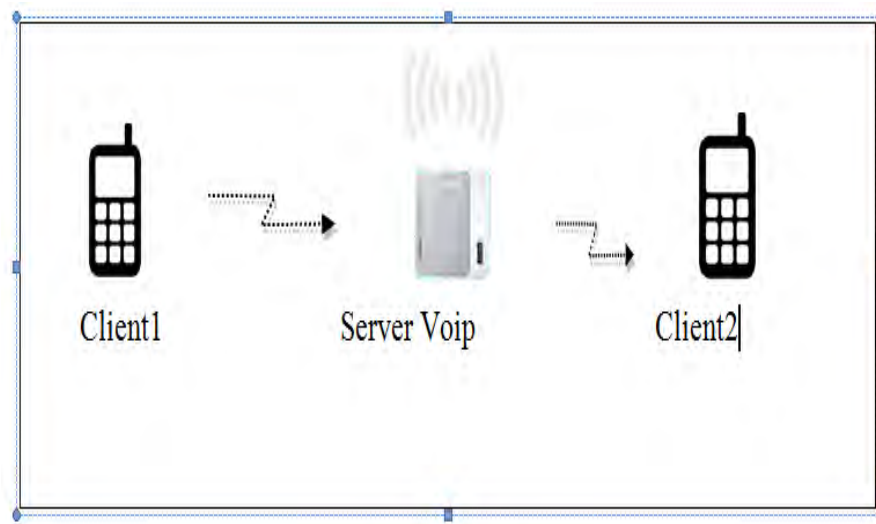
3.3.1 Pengujian Panggilan pada Sistem

Pengujian ini dilakukan untuk menguji kerja sistem yang sudah dibuat. Pengujian ini dilakukan dengan dua cara : pengujian pertama melakukan panggilan dari client-1 kepada client-2 tanpa jarak dan pengujian kedua dengan menentukan jarak panggilan.

a. Pengujian Sistem Server Voip

Pengujian dilakukan untuk mengetahui bahwa sistem dan konfigurasi yang dibuat pada server voip berjalan sesuai dengan rancangan. Pada gambar 3.3 menunjukkan pengujian dilakukan dengan cara melakukan panggilan dari client-1 kepada client-2 tanpa jarak. Pengujian sistem dinyatakan berhasil apabila client-1 dapat memanggil dan client-2 dapat menerima panggilan, serta dapat melakukan percakapan. Pengujian sistem tidak berhasil apabila :

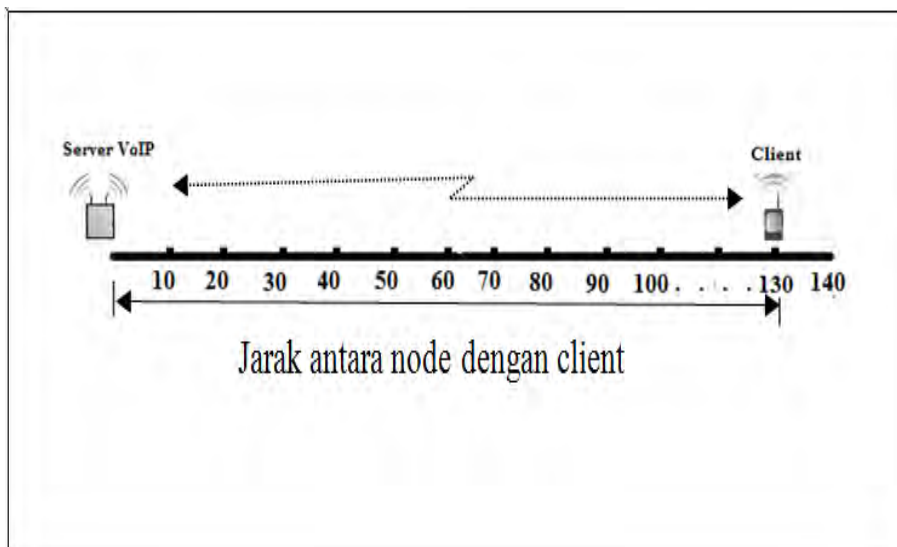
1. Signal panggilan terputus apabila client-2 menerima panggilan.
2. Salah satu client tidak bisa menerima panggilan.
3. Salah satu client tidak terdengar suara percakapan.



Gambar 3.3 Call antar client

b. Pengujian dengan jarak panggilan

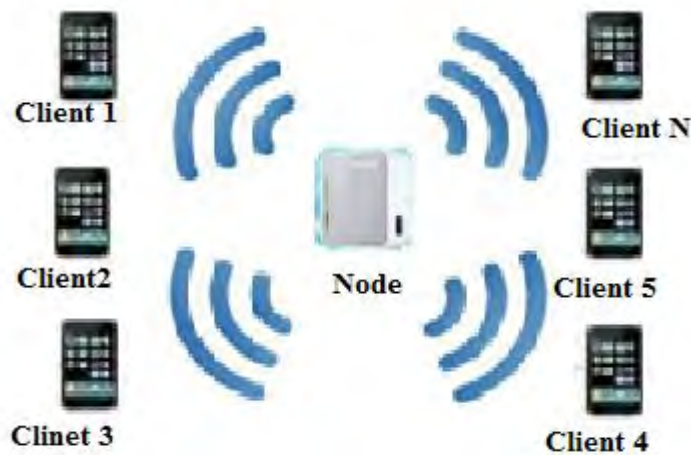
Pada pengujian ini akan dilakukan uji pemanggilan dari client1 ke client2. Jarak awal antara client1 dengan client2 adalah 10 meter, untuk jarak panggilan selanjutnya kelipatan 10 meter seperti terlihat pada gambar 3.4. Pengujian ini dilakukan untuk mencari kemampuan jarak terjauh yang dapat di cover node. Sehingga dapat diperkirakan jarak penempatan tiap node pada area implementasi.



Gambar 3.4 Jarak panggilan antara node dengan client

3.3.2 Pengujian Kapasitas Maksimum Panggilan

Pengujian ini berguna untuk mengetahui kapasitas layanan maksimum server VoIP sebagai node, sehingga dapat memperkirakan jumlah node yang harus dipasang pada saat implementasi nanti diilustrasikan pada gambar 2.5. Pengujian dilakukan dengan panggilan sebanyak mungkin secara simultan, dengan pengaturan waktu tertentu tiap panggilan. Di mana pemanggilan dengan cara berpasangan, pasangan client pertama tetap pada kondisi terkoneksi, hingga server voip tidak dapat melayani panggilan atau adanya panggilan sebelumnya yang terputus kemudian dihitung total panggilan yang berhasil. Hasil dari pengujian ini merupakan kemampuan real time, dengan mengaplikasikan metode native bridging pada server voip. Data hasil pengujian akan dibandingkan dengan data hasil pengujian kapasitas panggilan server voip tidak menggunakan metode native bridging.

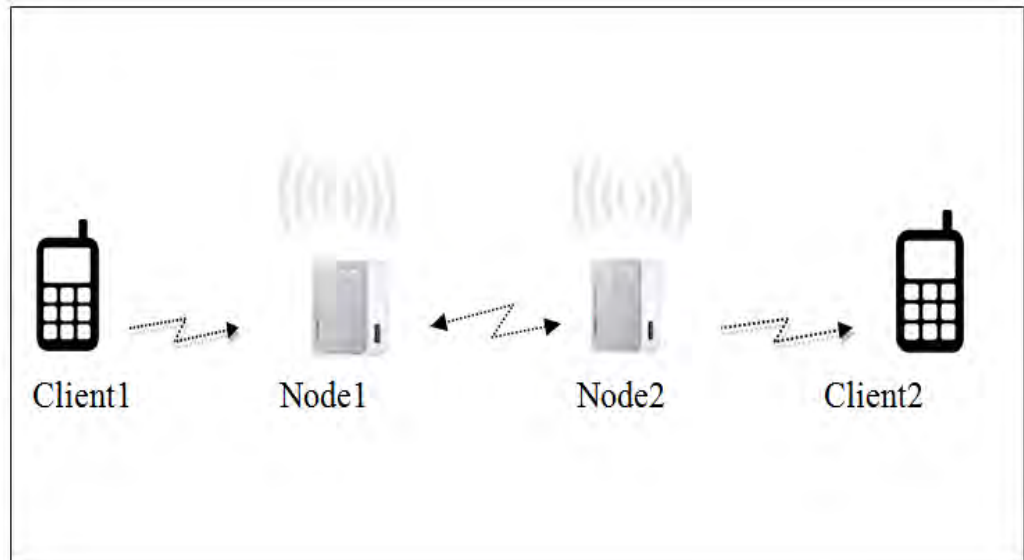


Gambar 3.5 Kapasitas Layanan per node

3.3.3 Pengujian Koneksi antar node

Pengujian ini dilakukan dengan cara menempatkan dua node atau lebih, di mana kondisi antar node tersebut saling terkoneksi, dan akan dilakukan pemanggilan dari client yang terkoneksi dengan node yang berbeda. Dari pengujian ini akan diketahui sistem bekerja dengan baik pada sistem jaringan, apabila antar client bisa memanggil dan menerima (saling terhubung) serta dapat melakukan percakapan.

Pada pengujian ini akan dilakukan juga pengukuran kapasitas panggilan yang mampu di layani oleh node server pada kondisi terkoneksi dengan node lain. Berikut gambar 3.6, model dari pengujian antar node



Gambar 3.6 Koneksi antar 2 Node

BAB 4

HASIL DAN PEMBAHASAN

4.1 Implementasi Rancangan

Pada bab ini dilakukan implementasi rancangan sistem, dengan tahapan instalasi dan konfigurasi sistem untuk membangun sebuah Server VoIP.

4.1.1 Instalasi Sistem OpenWRT pada Router

Setelah ditentukan perangkat, tahapan awal implementasi rancangan adalah menginstal firmware OpenWRT Barrier Beker 1.4 sebagai framework komunikasi berbasis embeded sistem yang di implementasikan pada Wireless VoIP.

Pada tahapan ini akan dilakukan instalasi firmware ke dalam device yang ditentukan. Langkah-langkah instalasinya sebagai berikut :

- a. Siapkan file firmware openwrt Barrier Beker 1.4, file ini bisa di download bebas di internet.
- b. Masuk kedalam halaman administrator dari device yang kita gunakan melalui browser. Pada kasus ini halaman administrator dapat diakses di <http://192.168.0.254>, setelah kita masukkan username dan passwordnya maka kita akan mendapati tampilan seperti pada gambar 4.1



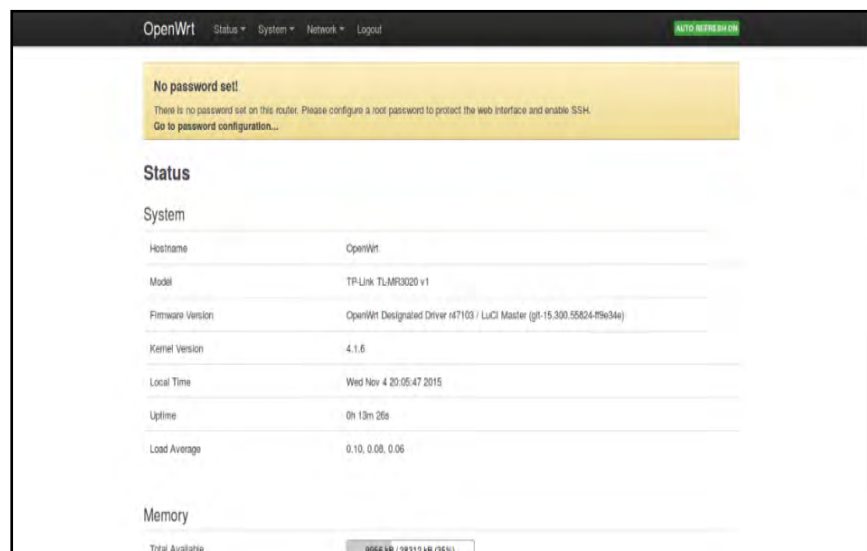
Gambar 4.1 Login router MR-3020

- c. Pilih menu sistem tools kemudian pilih firmware upgrade lihat gambar 4.2 dibawah. Pilihlah firmware openwrt Barrier Beker 1.4 yang sudah kita siapkan sebelumnya. Setelah semua selesai maka pilihlah upgrade. Prosesnya akan berjalan 5-10 menit.



Gambar 4.2 Proses Upgrade firmware OpenWRT

- d. Setelah selesai proses upgrade, lakukan konfigurasi IP pada AP/router menjadi 192.168.34.1 dan simpan konfigurasi tersebut, serta lakukan reboot
- e. Jika tidak ada masalah saat proses upgrade, maka openwrt bisa diakses melalui browser melalui dengan IP 192.168.34.1. tampak pada gambar 4.3 .



Gambar 4.3 Tampilan awal firmware

4.1.2 Konfigurasi Open Voice Sistem

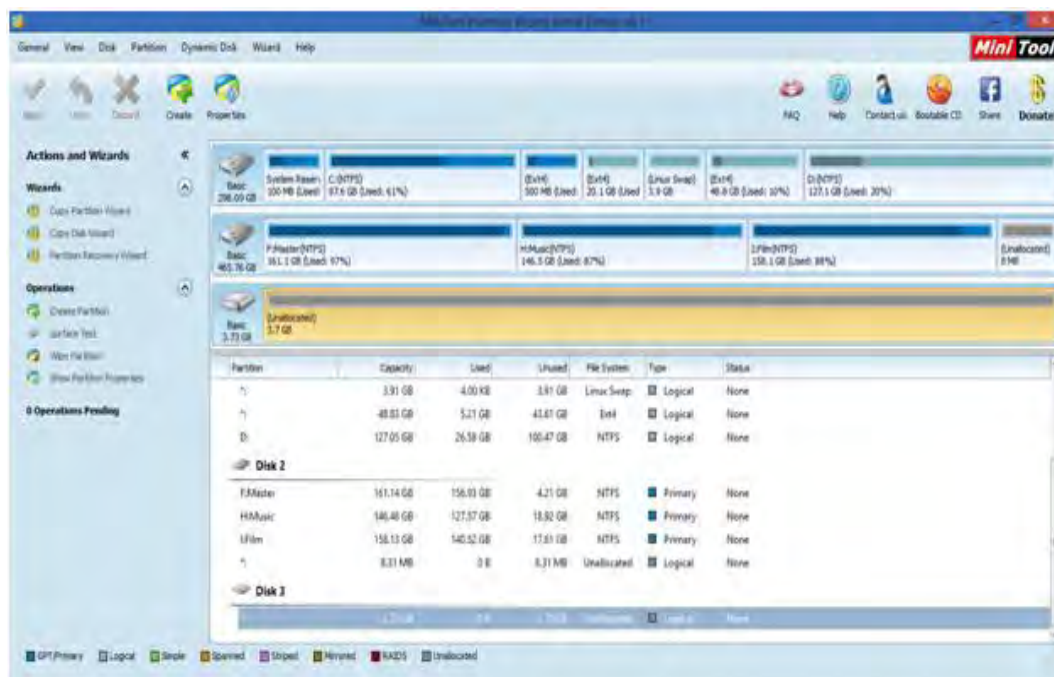
Untuk membuat sistem komunikasi yang berbasis Open-Voice dengan mengimplementasikan wireless VoIP, dan MANET maka diperlukan konfigurasi modul-modul yang sudah tertanam pada embedded device (router). Konfigurasi protokol pada sistem ini menggunakan OLSR sebagai routing protokolnya yang akan menghubungkan node-node yang ada dalam jaringan MANet. Serta konfigurasi codec yang di pilih jenis codec GSM

4.1.3 Install dan Konfigurasi Exroot

Pada proses konfigurasi ini diperlukan sebuah media storage flashdisk dengan kapasitas 16 GB, dengan kapasitas tersebut menjadikan lebih leluasa yang digunakan untuk menambah storage dan menyimpan modul-modul pada proses instalasi Asteris. Berikut tahapan proses dari exroot :

1. Proses Partisi Flashdisk

Sebelum dilakukan tahapan untuk exroot, terlebih dulu dilakukan partisi terhadap flashdisk menjadi dua bagian. Yaitu : partisi pertama untuk *swap* dengan kapasitas sebesar 15GB dan partisi kedua sebagai *root* dengan kapasitas 1GB. Berikut gambar 4.4, menunjukkan proses partisi.



Gambar 4.4 Partisi Flashdisk

2. Proses login dan proses konfigurasi Exroot

Untuk memulai tahapan exroot, dilakukan login pada root. Gambar 4.5a. menunjukkan proses login pada level root. Langkah selanjutnya update dengan memberikan beberapa perintah pada proses exroot, ditunjukkan pada gambar 4.5b, dibawah ini.



(a)

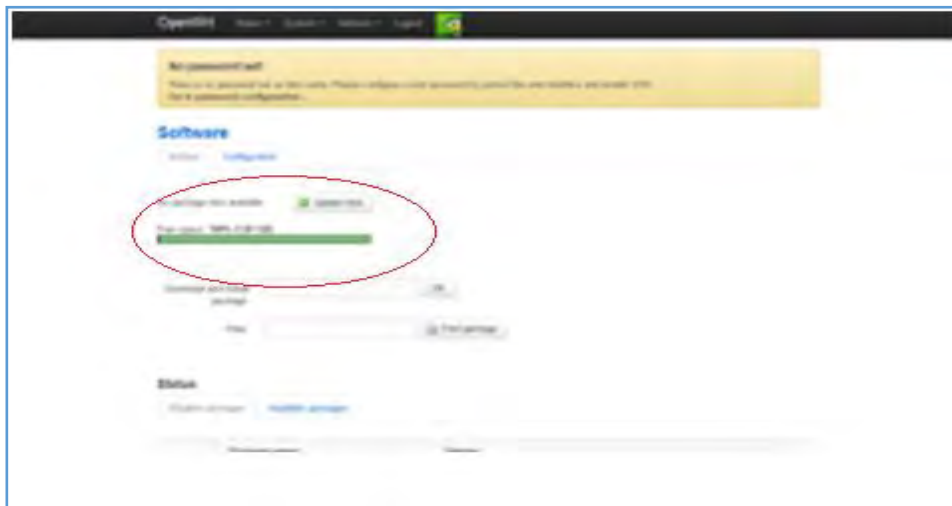


(b)

Gambar 4.5 (a)Login root dan (b) proses pemberian perintah exroot

3. Proses exroot berhasil

Berikut adalah proses akhir exroot yang ditunjukan oleh gambar 4.6 hasil exroot yang proses instalasi akan selesai (dengan tanda lingkaran warna merah)



Gambar 4.6 Proses akhir tahapan exroot

4.1.4 Instalasi Server VoIP

Pada tahapan ini dilakukan Instalasi asterisk pada sistem openwrt yang telah terembeded pada perangkat router pada tahapan sebelumnya diatas, sehingga perangkat router menjadi sebuah Server VoIP. Berikut adalah tahapan dalam instal asterisk yang perlu dilakukan.

- a) Langkah pertama adalah setup open instalasi asterisk beserta library codecnya yang akan dipakai dalam sistem. Instalasi dilakukan melalui service webbase yang dimiliki router ini agar lebih user friendly melalui repository offline yang telah dibuat sebelumnya untuk mempercepat proses instalasi paket.
- b) Setelah Instalasi selesai, langkah selanjutnya adalah konfigurasi asterisk yang berada di `/etc/asterisk` menggunakan aplikasi *winscp*
- c) Tahapan selanjutnya akan dilakukan konfigurasi OLSR routing protocol yang berfungsi sebagai protocol utama dalam mengatur aliran data. Karena dalam firmware ini sudah diikutkan dengan preconfigure file dari olsrd pada saat proses kompilasi firmware maka yang perlu dilakukan hanya menyesuaikan dengan ip address yang digunakan oleh interface yang digunakan.

4.1.5 Konfigurasi Native Bridging

Tahapan konfigurasi ini untuk mengubah mode awal Generic Bridging menjadi Native Bridging, dengan melakukan konfigurasi setting value pada dua file yaitu *sip.conf* dan *extension.conf*.

File Sip.conf memuat variabel Canreivite dengan nilai “Yes/No”, ini akan berfungsi untuk memotong atau lalu lintas RTP paket suara tidak melalui inti asterisk

File Extension.conf adalah file yang berisi aturan routing panggilan yang terjadi, jika asterisk akan melayani pengguna. Konfigurasi lengkap yang digunakan dalam penelitian ini terlampir.

Berikut dibawah ini adalah potongan konfigurasi native bridging pada file sip.conf dan Extension.conf.

a. **File Sip.Conf :**

```
[001]
type=friend
context=my-phones
username=001
secret=001
host=dynamic
nat=no ;
canreivite= yes ; memungkinkan RTP lalu lintas suara untuk
; memotong Asterisk

disallow=all
allow=gsm ; setting codes GSM
dtmfmode=info
```

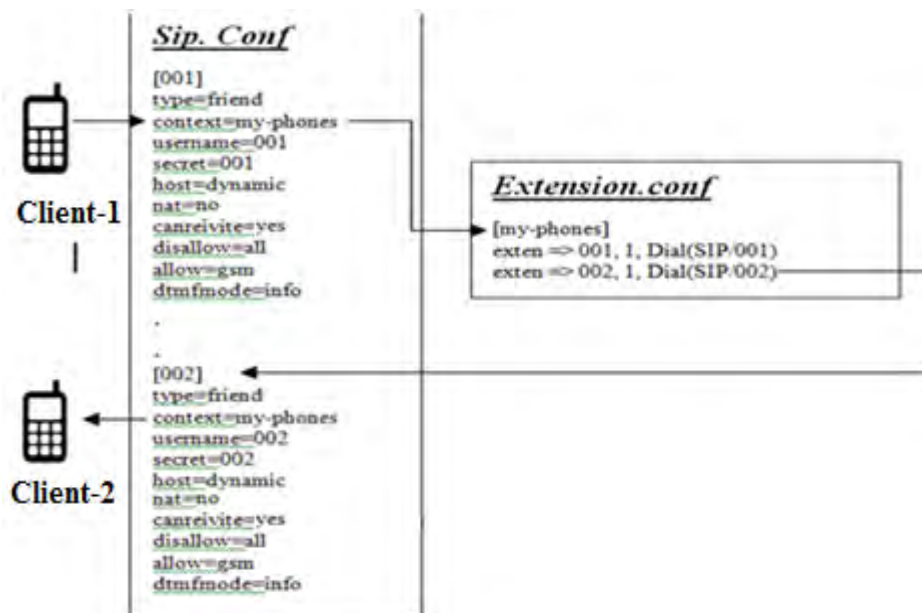
```
[002]
type=friend
context=my-phones
username=002
secret=002
host=dynamic
nat=no ;
canreivite= yes ; memungkinkan RTP lalu lintas suara untuk
; memotong Asterisk ;

disallow=all
allow=gsm ; setting codes GSM;
dtmfmode=info
```

b. **File extesion.conf:**

```
[my-phones]
exten => 001, 1, Dial(SIP/001)
exten => 002, 1, Dial(SIP/002)
```

.konfigurasi tersebut diatas, memberikan nilai *Canreivite= yes* ; dan *allow=gsm* , dengan konfigurasi tersebut diatas digunakan ketika kondisi reINVITEs yang diaktifkan. Dengan konfigurasi native bridging ini, audio mengalir di luar inti Asterisk pada server voip. sehingga endpoint mengirimkan media mereka langsung satu sama lain antar endpoint. Dengan syarat antara client telah diseting dengan codec yang sama yaitu GSM. Gambar 4.7 , memperlihatkan urutan proses pemanggilan antar client 001 ke client 002. Dengan konfigurasi native bridging.



Gambar 4.7 Urutan proses pemanggilan

4.2 Data Hasil Pengujian

Data hasil pengujian dan pengukuran baik itu berupa angka pada tabel dan data capture hasil pengujian yang telah dilakukan sesuai dengan bahasan pada bab sebelumnya Pengujian dan pengukuran dengan menggunakan software analyzer Wireshark version 1.12.4

4.2.1 Data Pengujian panggilan pada Sistem

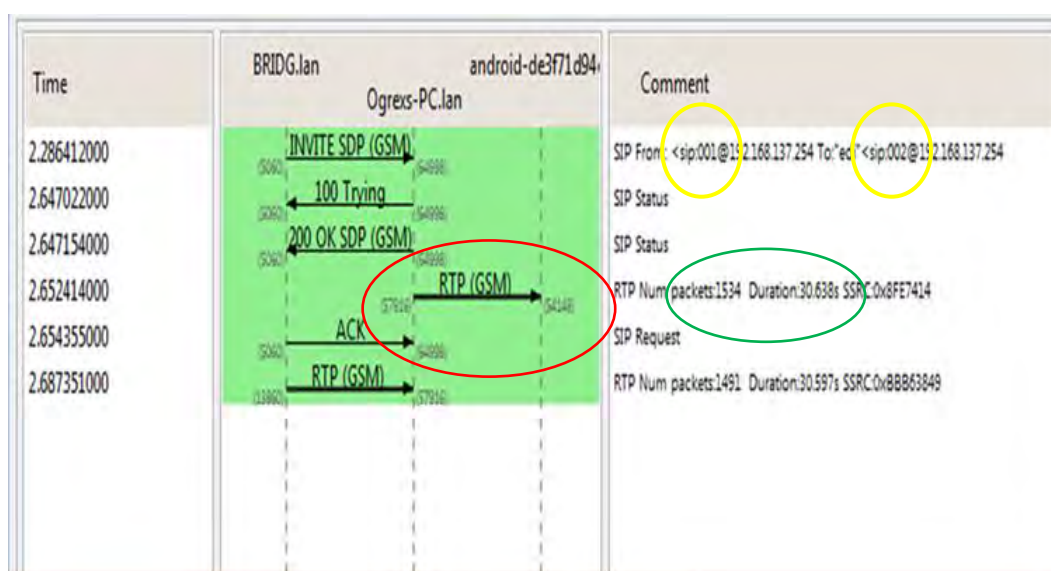
Pada pengujian dan pengukuran panggilan pada sistem, dilakukan dua pengujian, yaitu pengujian pada sisten server voip dan pengujian server voip atau node dengan penentuan jarak panggilan

a) Pengujian Sistem Server VoIP

Langkah awal pengujian sistem server voip, dengan cara mengkoneksikan antar client-1 dan client-2 ke node server voip, cara mengkoneksikan wifi pada smartphone masing client ke wifi node server

Untuk memastikan kedua client terkoneksi ke server/node, dapat dilihat pada server melalui fasilitas list network pada editor sisten openWRT. menunjukan beberapa client sudah terkoneksi dengan Server dan diantaranya sudah mendapatkan IP dari server voip.

Langkah berikutnya dilakukan pengujian dengan melakukan panggilan dari client-1 ke client-2. Gambar 4.8 menunjukkan panggilan dapat diterima dan terjadi percakapan antara client-1(sip001) sebagai pemanggil dengan client-2 (sip002) sebagai penerima (dengan tanda lingkaran warna kuning). Terjadinya penerimaan panggilan dan percakapan (dengan tanda lingkaran warna merah) dengan durasi 30,638 detik serta 1534 paket terkirim (dengan tanda lingkaran warna hijau), sebagai petunjuk konfigurasi native bridging berjalan dengan baik.



Gambar 4.8 Pengujian call Client-001 ke Client-002

b) Pengujian dengan jarak panggilan

Dari pengukuran jarak antara node dengan client didapatkan data seperti yang tertera pada Tabel 4.1. dimana telah dilakukan pengujian sekaligus pengukuran pada jarak awal 10 meter, dengan kelipantanya hingga jarak 100 meter lebih. Sehingga diperoleh selain jarak sinyal yang dapat dijangkau untuk menentukan area cakupan, juga diukur kualitas nilai throughput, delay, dan paket lose serta waktunya.

Tabel 4.1 Hasil pengukuran jarak antara node dengan client

Jarak (mtr)	Throughput(bps)	Delay(ms)	Paket loss (%)	Waktu (menit)
10	0,032	49,334	0	00,49
20	0,003	48,172	0	00,48
30	0,024	64,399	0	01,09
40	0,006	30,566	0	00,30
50	0,004	37,741	0	00,37
60	0,029	46,207	0	00,46
70	0,029	46,197	0	00,46
80	0,041	36,473	0	00,36
90	0,019	46,504	0	00,46
100	0,018	177,696	0	02,57
100+	0,004	66,096	-400%	01,06

4.2.2 Data Pengujian Kapasitas Maksimum Panggilan Per Node

a. Pengujian dan Pengukuran kapasitas layanan tanpa native bridging

Berikut adalah data hasil pengukuran kapasitas yang bisa dilayani per node dengan konfigurasi tidak menggunakan konfigurasi native bridging. Tabel 4.2 terlihat jumlah maksimal pasangan client yang mampu di layani oleh server voip, berjumlah 11 pasang atau 22 client.

Tabel 4.2 Data kapasitas layanan per node dengan generic bridging

Urutan Client	Throughput (Mbps)	Delay(ms)	Packet Loss %	Codec	Durasi Panggilan (m)
1	0,034	344,035	0	GSM FR	01:19
2	0,034	358,535	0	GSM FR	03:18
3	0,033	382,277	0	GSM FR	05:06
4	0,033	383,178	0	GSM FR	06:57
5	0,032	384,394	0	GSM FR	07:15
6	0,032	393,190	0	GSM FR	08:29
7	0,032	393,976	0	GSM FR	09:19
8	0,032	353,813	0	GSM FR	10:45
9	0,032	737,119	0	GSM FR	12:25
10	0,031	1,028,352	0,7	GSM FR	13:07
11	0,030	1,122,670	1.3	GSM FR	15:12

b. Pengujian dan pengukuran dengan konfigurasi Native Bridging.

Pada tabel 4.3 merupakan data hasil pengujian dan pengukuran dengan konfigurasi native bridging, jumlah client yang berhasil melakukan pemanggilan dan percakapan adalah sebanyak 13 pasang, ini menunjukkan jumlah pasangan maksimal panggilan yang dapat di dilayani oleh server voip. Dan apabila dilakukan pemanggilan oleh pasangan client berikutnya ke14 panggilan akan dilayani, tetapi dari salah satu pasangan client pemanggilan sebelumnya terputus. Ini menunjukkan kemampuan server voip maksimal melayani panggilan sebanyak 13 pasang client (26 user).

Tabel 4.3 Hasil pegujian kapasitas layanan per node dengan native bridging

Urutan Client	Throughput (Mbps)	Delay (ms)	Pkt Loss (%s)	Codec	Waktu (menit)
1	0,062	86,008	0	GSM FR	01:26
2	0,051	324,124	0	GSM FR	03:24
3	0,049	353,694	0	GSM FR	05:54

4	0,049	395,243	0	GSM FR	06:36
5	0,049	455,693	0	GSM FR	07:47
6	0,048	497,623	0	GSM FR	08:16
7	0,048	567,793	0	GSM FR	09:28
8	0,047	637,953	0	GSM FR	10:38
9	0,046	739,383	0	GSM FR	12:20
10	0,046	764,155	0	GSM FR	13:45
11	0,045	853,341	0	GSM FR	15:24
12	0,046	802,232	0	GSM FR	16:44
13	0,033	879,109	0	GSM FR	17:35

4.2.3 Data Pengujian Koneksi dan Kapasitas antar Node

Pada pengujian koneksi antar node dilakukan dua pengukuran, pengujian pertama adalah koneksi antar node dan pengujian kedua mengukur kapasitas panggilan antar node, berikut adalah tabel data :

- a. Dari pengujian koneksi antar node,

dan dilakukan pemanggilan dilanjutkan dengan percakapan antar client yang terkoneksi di masing-masing node, dilakukan sebanyak 3 kali pengujian, dengan hasil pengukuran tertera pada tabel 4.4, dari tabel tersebut ditunjukkan nilai throughput dari server maupun client, nilai paket loss serta jitter.

Tabel 4.4. Hasil pengukuran koneksi antar node (routing)

pengujian	Throughput (bps)		Paket Loss (%)	Jitter (ms)	Waktu (menit)
	Server	Client			
1	209,826	329.00	0	20	02:47
2	290,368	177.00	0	20	02:54
3	258,420	145.00	0	21	02:16

b. Pengujian kapasitas panggilan antar node

Pengujian kapasitas panggilan antar node dengan menambah node-2 pada sisten jaringan. Hasil pengukuran kapasitas yang dapat dilayani oleh server sejumlah 10 pasang client atau 20 client. Tabel 4.5 memaparkan hasil pengukuran jumlah client, nilai throughput, nilai delay dan nilai paket loss.

Tabel 4.5 Data pengukuran kapasitas panggilan antar node

Urutan Client	Throughput (Mbps)	Delay (ms)	Paket Loss (%)	Payload /Codec	Waktu (s)
1	0,047	354,36	0	GSM FR	01:39
2	0,048	330,99	0	GSM FR	02:27
3	0,047	434,37	0	GSM FR	03:21
4	0,046	507,45	0	GSM FR	04:28
5	0,045	700,59	0	GSM FR	05:41
6	0,045	824,16	0	GSM FR	07:35
7	0,046	514,62	0	GSM FR	08:13
8	0,043	964,59	0	GSM FR	09:42
9	0,032	1.021,56	0	GSM FR	10:39
10	0,025	1.450,26	0	GSM FR	12:09

4.3 Analisa Data Pengujian

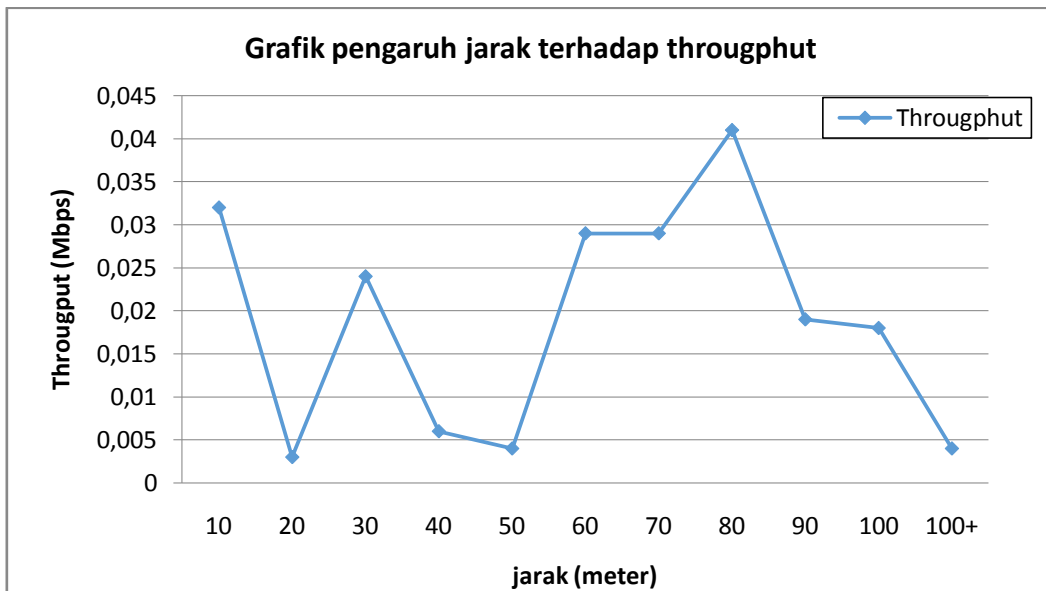
Data hasil pengukuran pada pengujian ini akan ditampilkan dalam bentuk format grafik, untuk memudahkan analisa kualitas VoIP.

4.3.1 Analisa Data Pengujian Call pada Sistem

Untuk analisa data hasil pengukuran dan pengujian pemanggilan terhadap sistem terdapat 3 variabel pengukuran, yaitu :

a) *Throughput*

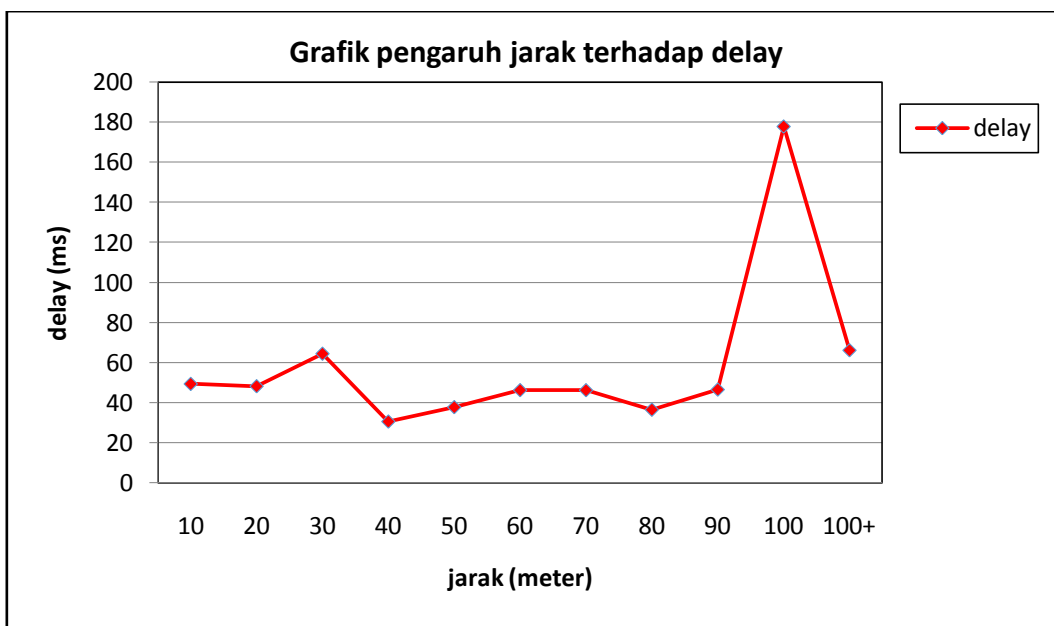
Grafik gambar 4.2 menunjukkan nilai throughput yang tidak stabil antara jarak pengukuran 10-60 meter. Pada jarak 80 meter nilai throughput pada titik tertinggi. Pada jarak 90-100 meter lebih nilai throughput melemah, disebabkan daya tangkap sinyal node semakin melemah karena jarak semakin jauh.



Gambar 4.9 Grafik pengaruh jarak terhadap troughput

b) Delay

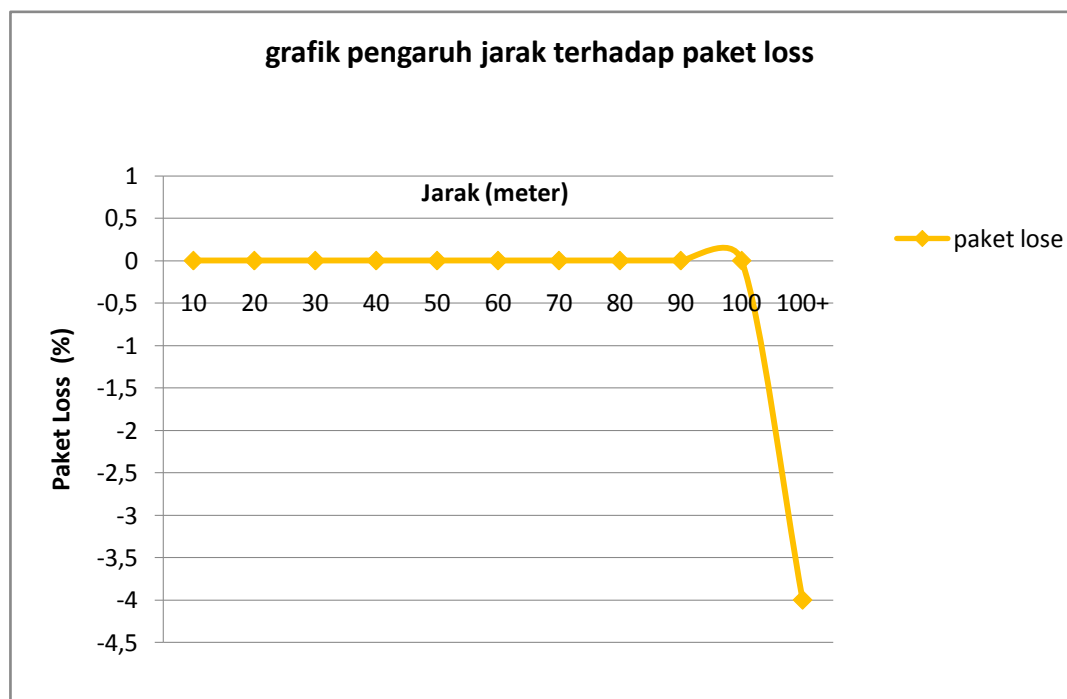
Terlihat pada gambar 4.10 bahwa secara umum nilai delay yang diterima memiliki trend yang cenderung meningkat, berbanding lurus dengan pertambahan jarak yang terjadi. Nilai delay tertinggi pada posisi jarak 100 meter. Hal ini karena semakin lamanya waktu yang diperlukan untuk melakukan pengiriman data yang menyebabkan delay yang semakin tinggi pula..



Gambar 4.10 Grafik pengaruh jarak terhadap delay

c) Paket Loss

Grafik yang ditampilkan pada gambar 4.11, dapat dilihat bahwa nilai paket loss cenderung stabil pada jarak pengukuran 10 meter sampai 90 meter, dan pada jarak melebihi 100 meter, paket loss berkurang secara signifikan hal ini terjadi karena dipengaruhi nilai delay turun



Gambar 4.11 Grafik pengaruh jarak terhadap paket loss

4.3.2 Analisa Data Kapasitas Maksimum Panggilan Per Node

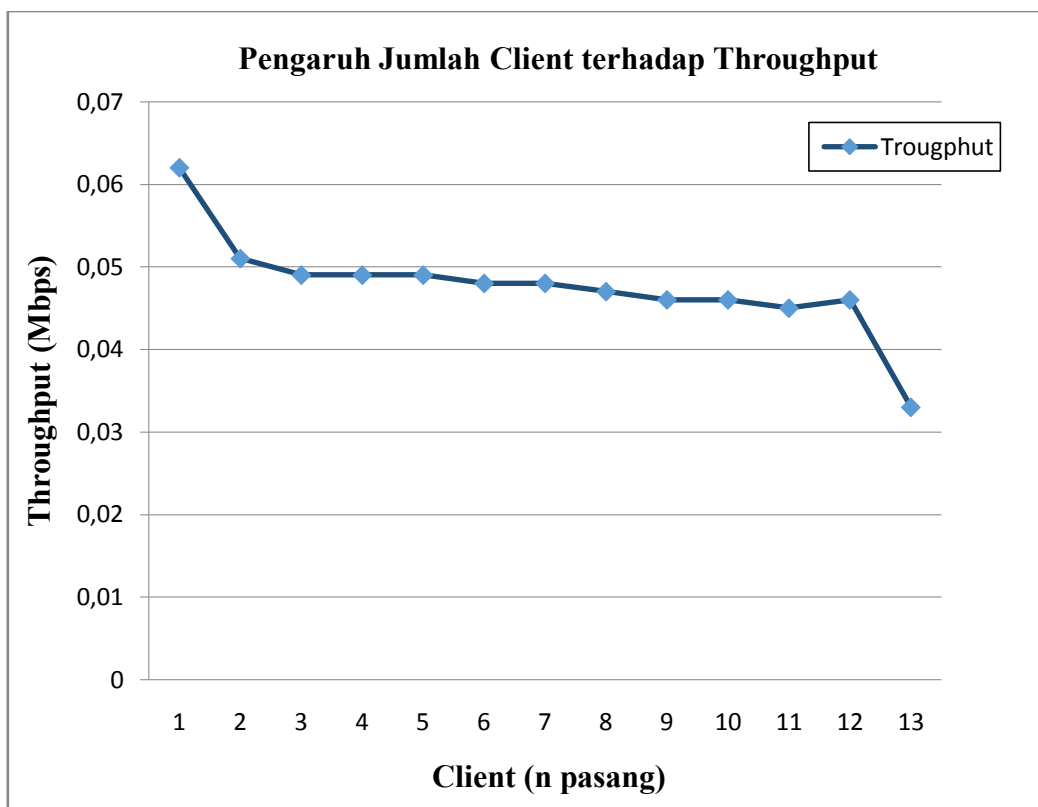
Pada analisa data kapasitas panggilan terbagi menjadi dua, yaitu analisa data hasil pengukuran kapasitas panggilan dengan konfigurasi native bridging dan analisa perbandingan antara data kapasitas panggilan dengan metode native dan generig bridging.

Berikut adalah analisa meliputi kualitas nilai throughput dan delay dari data hasil pengukuran kapasitas panggilan dengan konfigurasi native bridging.

a) Throughput

Pada grafik gambar 4.12, menunjukkan pasangan terakhir yang melakukan aktifitas pemanggilan adalah pasangan client-13. Saat pasangan selanjutnya

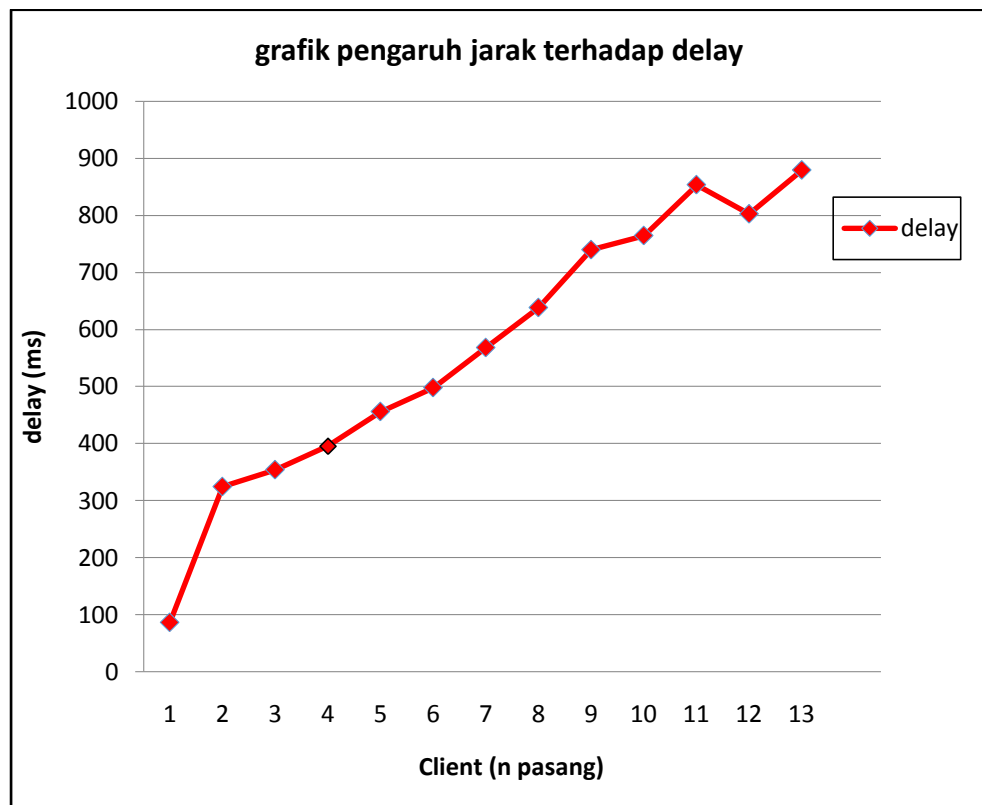
client-14 melakukan panggilan masuk ke dalam sistem mengakibatkan salah satu pengguna sebelumnya terputus dari sistem. Hal ini dikarenakan dengan bertambahnya pengguna yang terkoneksi ke dalam sistem maka semakin berkurang juga throughput yang di dapat. sehingga berdampak pada besar data yang bisa dikirimkan berkurang. Dan dapat di analisa dalam pengujian didapatkan hasil bahwa dengan bertambahnya client, menjadikan throughput yang diperoleh cenderung semakin kecil dikarenakan berbagi saluran.



Gambar 4.12 Pengaruh jumlah client terhadap throughput

b) Delay

Grafik yang tersaji pada gambar 4.13, dapat dilihat bahwa nilai delay mengalami kecendrungan meningkat, ini berbanding lurus dengan pertambahan jumlah client yang melakukan pemanggilan. Hal ini dikarenakan dengan bertambahnya jumlah aktifitas pemanggilan, maka delay cenderung akan semakin besar.



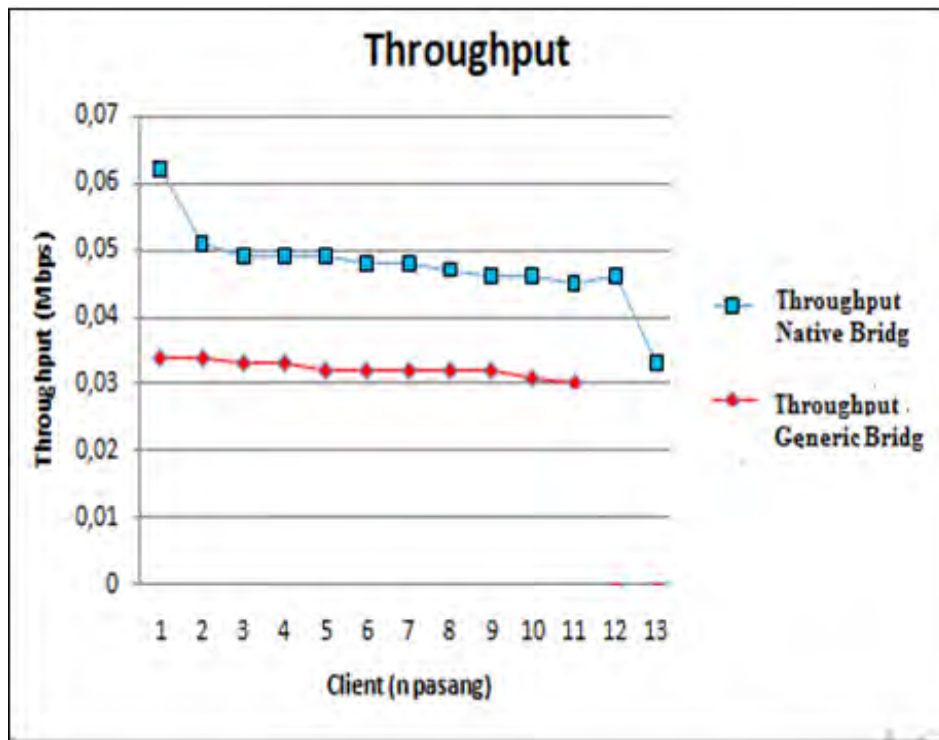
Gambar 4.13 Pengaruh jumlah client terhadap delay

Berikut adalah analisa perbandingan antara data kapasitas panggilan dengan metode native bridging dan generic bridging. Analisa perbandingan ini selain membandingkan jumlah kapasitas panggilan, juga dianalisa kualitas nilai throughput dan delay pada kedua konfigurasi.

Jumlah maksimal kapasitas panggilan pada native bridging berjumlah 26 client (13 pasang), sedangkan jumlah maksimal kapasitas panggilan pada generic bridging berjumlah 22 client (11 pasang). Terjadi peningkatan sebesar 18,18% , berikut untuk analisa pada kualitas troughput dan delay.

a. Throughput

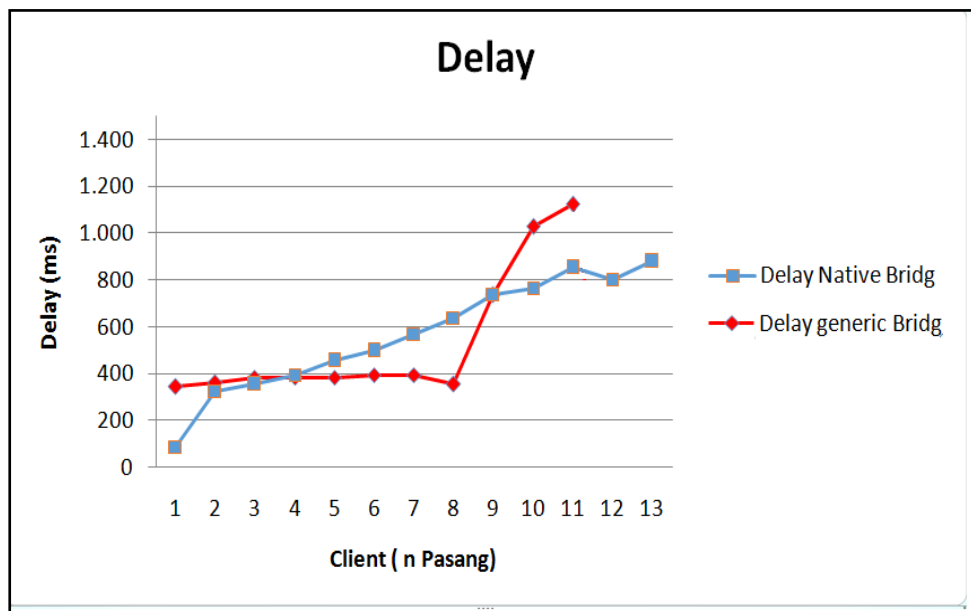
Pada gambar 4.14. grafik terlihat dengan jumlah panggilan yang sama, nilai throughput untuk metode native bridging lebih tinggi, dengan nilai diatas 0,04 Mbps dibandingkan dengan nilai throughput untuk generic bridging yang rata-rata dibawah 0.040 Mbps. Ini menunjukkan dengan menggunakan metode native bridging terjadi peningkatan nilai throughput pada sistem komunikasi voip yang digunakan.



Gambar 4.14 Perbandingan Throughput pada native dan generic bridging

b. Delay

Grafik nilai delay pada gambar 4.15, menunjukkan nilai delay baik itu untuk native maupun generic bridging sama-sama mengalami peningkatan. Tetapi delay pada native bridging diawali dengan nilai delay yang kecil yaitu 86,08 ms dan seiring dengan jumlah client yang masuk, delay mengalami kenaikan tapi stabil. Sedangkan delay pada generic bridging diawali dengan nilai di atas 300 ms dan terjadi peningkatan yang signifikan hingga 1000 ms pada saat masuknya client ke 8 sampai terakhir. Jadi pada saat urutan sama-sama client ke 8, delay pada konfigurasi native bridging lebih rendah dibanding dengan delay pada konfigurasi generic bridging.



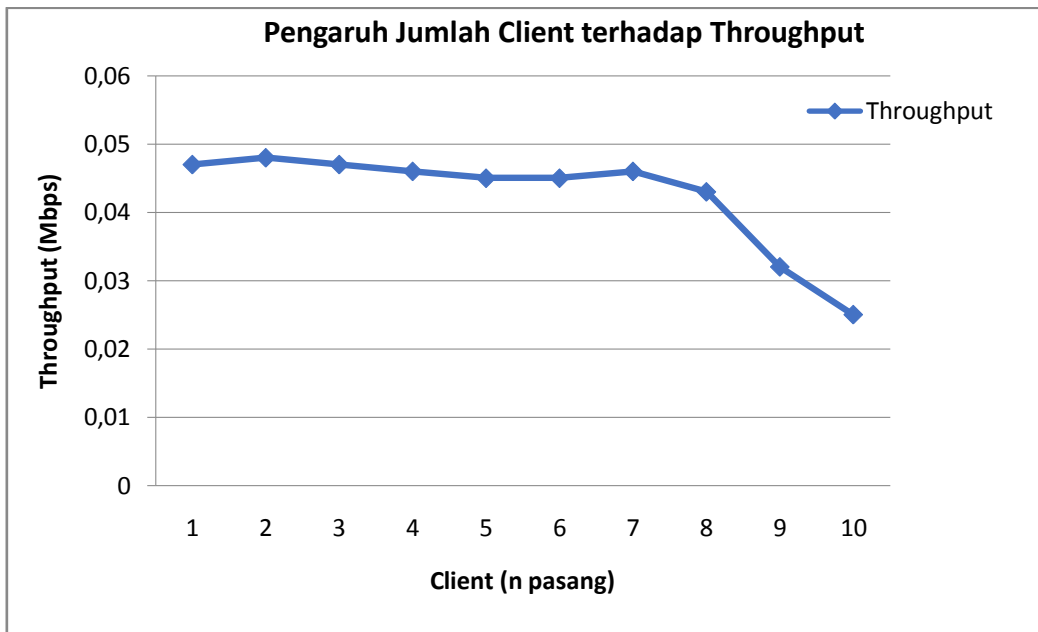
Gambar 4.15 Perbandingan Delay pada native dan generic bridging

4.3.3 Analisa Data Pengujian Koneksi dan Kapasitas Panggilan Antar Node

Pada analisa data koneksi antar node terbagi menjadi dua yaitu analisa data hasil pengukuran koneksi antar node dan analisa kapasitas panggilan antar node. Berikut ini adalah analisa data hasil dari pengukuran kapasitas panggilan antar node, yang meliputi nilai kualitas throughput dan delay :

a. Throughput

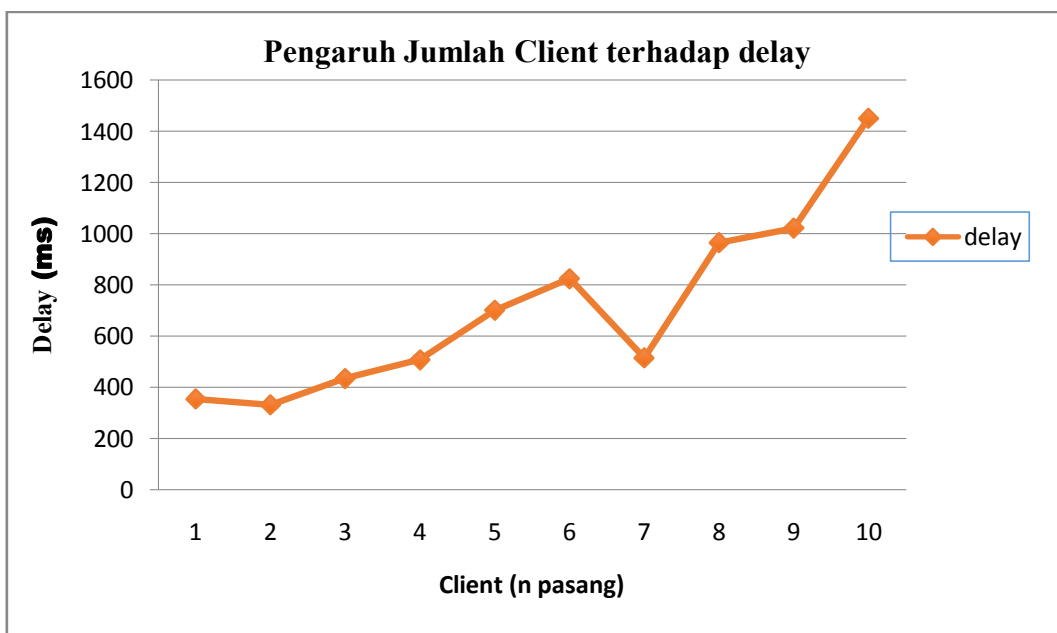
Pada grafik gambar 4.16, menunjukkan pasangan terakhir yang melakukan aktifitas pemanggilan adalah pasangan client-10. Saat pasangan selanjutnya client-11 melakukan panggilan masuk kedalam sistem mengakibatkan salah satu pengguna sebelumnya terputus dari sistem. Hal ini dikarenakan dengan bertambahnya pengguna yang terkoneksi kedalam sistem maka semakin berkurang juga throughput yang di dapat. sehingga berdampak pada besar data yang bisa dikirimkan berkurang. Dan dapat di analisa dalam pengujian didapatkan hasil bahwa dengan bertambahnya client, menjadikan throughput yang diperoleh cenderung semakin kecil dikarenakan berbagi saluran.



Gambar 4.16 Throughput pada kapasitas panggilan antar node

b. Delay

Grafik yang tersaji pada gambar 4.17, dapat dilihat bahwa nilai delay mengalami kecenderungan meningkat, ini berbanding lurus dengan pertambahan jumlah client yang melakukan pemanggilan. Hal ini dikarenakan dengan bertambahnya jumlah aktifitas pemanggilan, maka delay cenderung akan semakin besar



Gambar 4.17 Delay pada kapasitas panggilan antar node

!-- Halaman ini sengaja di kosongkan --!

HALAMAN LAMPIRAN

7.1 Konfigurasi Penambahan Kapasitas memori

----- Extroot Configuration -----

```
#opkg install block-mount kmod-fs-ext4 kmod-usb-storage kmod-usb-ohci
#reboot
#block info
#mount /dev/sda1 /mnt
#mkdir /tmp/cproot
#mount --bind / /tmp/cproot
#tar -C /tmp/cproot -cvf - . | tar -C /mnt -xvf -
#sync ; umount /mnt
#umount /tmp/cproot
#block detect > /etc/config/fstab
#vi /etc/config/fstab
#/ect/init.d/fstab enable
#/etc/init.d/fstab start
#reboot
#df -h
```

7.2. Konfigurasi Native Bridging

File Sip. Conf

.....

[001]

```
type=friend
context=my-phones
username=001
secret=001
host=dynamic
nat=no
canreinvite= yes
disallow=all
allow=gsm
dtmfmode=info
```

; memungkinkan RTP lalu lintas suara untuk

[002]

```
type=friend
context=my-phones
username=002
```

secret=002
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[003]
type=friend
context=my-phones
username=003
secret=003
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[004]
type=friend
context=my-phones
username=004
secret=004
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[005]
type=friend
context=my-phones
username=005
secret=005
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[006]
type=friend
context=my-phones
username=006
secret=006
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[007]
type=friend
context=my-phones
username=007
secret=007
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[008]
type=friend
context=my-phones
username=008
secret=008
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[009]
type=friend
context=my-phones
username=009
secret=009
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk

```
disallow=all  
allow=gsm  
dtmfmode=info
```

```
[010]  
type=friend  
context=my-phones  
username=010  
secret=010  
host=dynamic  
nat=no  
canreinvite=yes ; memungkinkan RTP lalu lintas suara untuk  
disallow=all  
allow=gsm  
dtmfmode=info
```

```
[011]  
type=friend  
context=my-phones  
username=011  
secret=011  
host=dynamic  
nat=no  
canreinvite=yes ; memungkinkan RTP lalu lintas suara untuk  
disallow=all  
allow=gsm  
dtmfmode=info
```

```
[012]  
type=friend  
context=my-phones  
username=012  
secret=012  
host=dynamic  
nat=no  
canreinvite=yes ; memungkinkan RTP lalu lintas suara untuk  
disallow=all  
allow=gsm  
dtmfmode=info
```

```
[013]  
type=friend  
context=my-phones  
username=013
```

secret=013
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[014]
type=friend
context=my-phones
username=014
secret=014
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[015]
type=friend
context=my-phones
username=015
secret=015
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[016]
type=friend
context=my-phones
username=016
secret=016
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

```
[017]
type=friend
context=my-phones
username=017
secret=017
host=dynamic
nat=no
canreinvite= yes      ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info
```

```
[018]
type=friend
context=my-phones
username=018
secret=018
host=dynamic
nat=no
canreinvite= yes      ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info
```

```
[019]
type=friend
context=my-phones
username=019
secret=019
host=dynamic
```

```
[020]
type=friend
context=my-phones
username=020
secret=020
host=dynamic
nat=no
canreinvite= yes      ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info
```

[021]
type=friend
context=my-phones
username=021
secret=021
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[022]
type=friend
context=my-phones
username=022
secret=022
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[023]
type=friend
context=my-phones
username=023
secret=023
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[024]
type=friend
context=my-phones
username=024
secret=024
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk

disallow=all
allow=gsm
dtmfmode=info

[025]
type=friend
context=my-phones
username=025
secret=025
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[026]
type=friend
context=my-phones
username=026
secret=026
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[027]
type=friend
context=my-phones
username=027
secret=027
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[028]
type=friend
context=my-phones

username=028
secret=028
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[029]
type=friend
context=my-phones
username=029
secret=029
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

[030]
type=friend
context=my-phones
username=030
secret=030
host=dynamic
nat=no
canreinvite= yes ; memungkinkan RTP lalu lintas suara untuk
disallow=all
allow=gsm
dtmfmode=info

Extension.Conf

.....
[my-phones]
exten => 001, 1, Dial(SIP/001)
exten => 002, 1, Dial(SIP/002)
exten => 003, 1, Dial(SIP/003)
exten => 004, 1, Dial(SIP/004)
exten => 005, 1, Dial(SIP/005)
exten => 006, 1, Dial(SIP/006)
exten => 007, 1, Dial(SIP/007)
exten => 008, 1, Dial(SIP/008)

```

exten => 009, 1, Dial(SIP/009)
exten => 010, 1, Dial(SIP/010)
exten => 011, 1, Dial(SIP/011)
exten => 012, 1, Dial(SIP/012)
exten => 013, 1, Dial(SIP/013)
exten => 014, 1, Dial(SIP/014)
exten => 015, 1, Dial(SIP/015)
exten => 016, 1, Dial(SIP/016)
exten => 017, 1, Dial(SIP/017)
exten => 018, 1, Dial(SIP/018)
exten => 019, 1, Dial(SIP/019)
exten => 020, 1, Dial(SIP/020)
exten => 021, 1, Dial(SIP/021)
exten => 022, 1, Dial(SIP/022)
exten => 023, 1, Dial(SIP/023)
exten => 024, 1, Dial(SIP/024)
exten => 025, 1, Dial(SIP/025)
exten => 026, 1, Dial(SIP/026)
exten => 027, 1, Dial(SIP/027)
exten => 028, 1, Dial(SIP/028)
exten => 029, 1, Dial(SIP/029)
exten => 030, 1, Dial(SIP/030)

```

7.3. Konfigurasi OLSR

```

----- OLSR -----

config olsrd
option IpVersion '4'

config LoadPlugin
option library 'olsrd_arprefresh.so.0.1'

config LoadPlugin
option library 'olsrd_dyn_gw.so.0.5'

config LoadPlugin
option library 'olsrd_httpinfo.so.0.1'
option port '1978'
list Net '0.0.0.0 0.0.0.0'

config LoadPlugin
option library 'olsrd_nameservice.so.0.3'

config LoadPlugin
option library 'olsrd_txtinfo.so.0.1'
option accept '0.0.0.0'

```

```
config Interface
option ignore '0'
option interface 'mywifi'
option Mode 'mesh'
config InterfaceDefaults
```

```
config LoadPlugin
option library 'olsrd_bmf.so.1.7.0'
option ignore '1'
```

```
config LoadPlugin
option library 'olsrd_dot_draw.so.0.3'
option ignore '1'
```

```
config LoadPlugin
option library 'olsrd_dyn_gw_plain.so.0.4'
option ignore '1'
```

```
config LoadPlugin
option library 'olsrd_jsoninfo.so.0.0'
option ignore '1'
```

```
config LoadPlugin
option library 'olsrd_pgraph.so.1.1'
option ignore '1'
```

```
config LoadPlugin
option library 'olsrd_mdns.so.1.0.0'
option ignore '1'
```

```
config LoadPlugin
option library 'olsrd_p2pd.so.0.1.0'
option ignore '1'
```

```
config LoadPlugin
option library 'olsrd_quagga.so.0.2.2'
option ignore '1'
```

```
config LoadPlugin
option library 'olsrd_secure.so.0.6'
option ignore '1'
```

```
config LoadPlugin
option library 'olsrd_tas.so.0.1'
option ignore '1'
```

```
config LoadPlugin
```

```
option library 'olsrd_watchdog.so.0.1'  
option ignore '1'
```

```
config Hna4  
option netaddr '192.168.34.0'  
option netmask '255.255.255.0'
```